Connection & File Services

Benutzerhandbuch

Ausgabe Januar 2015 (CFS V18.4)

Vorwort

Was enthält dieses Manual ?

Das vorliegende Manual beschreibt das Programm CFS (Connection & File Services), das auf allen DV-Anlagen mit dem Betriebssystem BS2000 der Firma Siemens-Nixdorf ablauffähig ist.

Das Manual besteht aus folgenden Kapiteln:

1	**Einführung** Der Leser erhält eine Übersicht über die Funktionen von CFS, sowie eine Auswahl einfacher Anwendungsbeispiele für CFS.
2	**Begriffserläuterungen** In diesem Kapitel werden die in dieser Beschreibung verwendeten Metazeichen sowie die am häufigsten verwendeten Fachbegriffe erklärt.
3	**Die wichtigsten Bildschirmformate von CFS** Einführung in die Arbeitsweise mit dem Programm CFS anhand der verwendeten Bildschirmformate (Masken).
4 - 9	**Beschreibung der Eingabemöglichkeiten und Kommandos**
10 - 16	**Zusammenfassende Darstellung bestimmter Themenkreise.** Menue, Job-Report-System, File-Transfer, Datei-Informationssystem, ...
17	**Benutzerschnittstellen** Hier werden die Schnittstellen beschrieben, nach denen der Benutzer eigene Verarbeitungsroutinen in das Programm CFS integrieren kann.
18	**Parameter ändern** Beschreibung der Kommandos zur Änderung von CFS-internen Parametern. Durch diese Parameter kann das Programmverhalten beeinflußt werden.
19 - 21	**Prozedursprache, automatischer Startup, USERLIB** In diesen Kapiteln ist die CFS-eigene Prozedursprache sowie deren Anwendungsmöglichkeit für wiederkehrende Arbeitsabläufe beschrieben.
22 - 23	**Wiederanlauf im Fehlerfall, Terminierungsabfrage**
24	**Help-System** Beschreibung des Online-Hilfesystems in CFS. Über dieses System erhalten Sie auch am Bildschirm alle in diesem Manual abgedruckten Informationen.
25 - 27	**K-/F-/DUE-Tasten, Prozeßschalter, verwendete Linknamen usw.**
S	**Stichwörter**

Welche Vorkenntnisse sind nötig ?

Für die Arbeit mit dem Programm CFS benötigen Sie Grundkenntnisse des BS2000. Bei Aufruf von Dienstprogrammen wie EDT, ARCHIVE, PERCON, LMS werden Grundkenntnisse dieser Programme vorausgesetzt.

Wie finden Sie sich in diesem Manual zurecht ?

Über das Inhaltsverzeichnis können Sie ein Kommando, dessen Name Ihnen bereits bekannt ist, direkt aufsuchen. Das Inhaltsverzeichnis ist so aufgebaut, daß Sie den Namen einer Funktion auch aufgrund der angegebenen Beschreibung auffinden können.
Das Stichwortverzeichnis bietet darüber hinaus Querverweise über das ganze Manual zu allen CFS-spezifischen Begriffen.

In den Kapiteln 10 bis 16 finden Sie zusammengefaßte Erläuterungen zu bestimmten Themenkreisen.

Falls Sie Verbesserungsvorschläge oder Anregungen zum Benutzerhandbuch oder zu bestimmten Funktionen des Programms CFS haben, rufen Sie uns an oder schreiben Sie uns auf dem dafür vorgesehenen Formblatt.

OPG Online-Programmierung GmbH

Sendlinger Str. 28, D-80331 München
Tel. 089/267831

Im folgenden sind die Erweiterungen des Programms CFS seit Januar 2000 zusammengestellt.

04.02.2014 Die Variable Action ONXFIND(suchdatei) verarbeitet nun Suchdateien mit nahezu unbegrenzt vielen Suchargumenten. Die einzige Einschränkung ist die Größe des freien Klasse 6 Speichers im oberen Adressraum.

12.04.2013 Dateienlisten bzw. ZIP-Archive mit mehr als 21000 Einträgen sind nun auch unter Nicht-TSOS Kennungen möglich.

31.03.2013 Action-Code EDT für Elemente einer PLAM-Bibliothek: Mittels einer Modifikation im Benutzermodul CFSMAIN kann veranlasst werden, dass beim Action-Code EDT nur die Sätze mit der Satzart 1 angezeigt werden. Sätze mit anderen Satzarten werden von CFS zwischengespeichert und beim Action-Code UPD wieder hinzugemischt.

01.02.2013 Erweiterung der Editorkommandos IR und IL: Durch Eingabe von IL n+ bzw. IL n- werden die neuen Sätze mit der kleinstmöglichen Schrittweite oberhalb des unteren Schlüssels (+) bzw. unterhalb des oberen Schlüssels (-) eingefügt.

04.01.2013 Erweiterung des Action-Codes REO um REOL (Reorganisation der PLAM-Bibliothek mit Unterprogramm LMS).
SET SEARCHMODE=A: Zeichenfolgen im Suche-Kommando werden wie A'...' behandelt.
SET REPLACEMODE=L: Bei Treffern im Suche-Kommando S,'abc'*'def'='xyz' wird der letzte gefundene Teilstring durch 'xyz' ersetzt.
Überarbeitete Funktionalität der User Option SNAM

18.07.2012 Anpassungen aller privilegierten Module von CFS an OSD V9.0
In der Dateienliste wird mit dem Zeichen % vor dem Lastp angezeigt, daß die Datei auf einem NET-STORAGE Dateisystem gespeichert ist.
Mit der Eingabe NET im Feld VOLUME der Selektionsmaske werden alle Dateien selektiert, die auf einem NET-STORAGE Dateisystem gespeichert sind.

18.04.2012 Ab OSD V7: zusätzlicher Parameter PL bei User Option OPEN
Es werden alle geöffneten PLAM-Bibliotheken zusammen mit den darin geöffneten Elementen angezeigt.
Der Action-Code OPN bei einer PLAM-Bibliothek zeigt alle Tasks, die diese Bibliothek geöffnet haben zusammen mit den geöffneten Elementen.

09.12.2011 Im Display-Modus von CFS kann die Bildschirmanzeige mit SWS2/SWS3W/SWS4W auf 132 Spalten und 27 bis 43 Zeilen vergrößert werden.

02.02.2011 Unterstützung von ZIP-Archiven im BS2000:
Mit der Variablen Action ONXZIP können nun auch passwortgeschützte Archive angelegt werden. Mit ONXUNZIP können Elemente aus passwortgeschützten Archiven selektiert werden. Der entsprechende Zusatzparameter lautet: PASS=

07.10.2010 Neuer Parameter zum Festlegen der Einfügeposition für kopierte Dateien.
SET COPYMODE=A/E

15.09.2010 Überarbeitete Compare-Funktion.
Das Kommando COMP erfuhr ein grundlegendes Redesign bezüglich der Parameter und der Bildschirmmaske. Das alte COMP-Kommando kann noch aus Kompatibilitätsgründen mit COMP0 aufgerufen werden.

Änderungsprotokoll

01.06.2010	Inkompatible Änderung: Der CSO-Modus (Compress Screen Output) ist nun standardmäßig aktiviert und kann durch CFSMAIN oder das Kommando NCSO ausgeschaltet werden. Der CSO-Modus hat jetzt noch einen weiteren Effekt: Beim Suche-Kommando im Datei-Display werden die gefundenen Treffer Items am Bildschirm hell hervorgehoben dargestellt.
29.01.2010	Ab OSD V8.0 kann der Action-Code CKD (Copy Keep Date) auch unter Kennungen ungleich TSOS angegeben werden.
21.12.2009	Unter Nicht TSOS-Kennungen bewirkt die Eingabe () im Feld USER-ID der Selektionsmaske, daß eine Datei CFS.USERID.COOWNER gelesen wird. In der Datei sind alle User-IDs aufgeführt, für die die eigene Kennung die Coownerschaft besitzt. Damit werden alle entsprechenden User-IDs selektiert.
14.12.2009	Variable Action ONXCOPY *:pvs:* Die angekreuzten Dateien werden unter der gleichen Kennung auf den angegebenen Pubset kopiert.
15.07.2009	Als Standard Sort Option ist nun auch SO NAME(1,44) zulässig. Damit kann bei PLAM-Bibliotheken eine besondere Sortierung nach Elementnamen erreicht werden, wobei die verschiedenen Typen eines Elements nacheinander aufgelistet werden.
13.05.2009	Ab OSD V6.0 und unter TSOS: User Option OPEN RL (Record Lock) selektiert alle mit SHARUPD=YES eröffneten ISAM-Dateien, für die durch die angezeigte TSN eine Satz-/Blocksperre gehalten wird.
08.03.2009	Kommando SWS (Set Window Size) zum Festlegen der Zeilenanzahl in der Dateienliste von CFS: Im 9763-Modus stehen die Varianten SWS3/SWS4 (32/43 Zeilen) zur Verfügung.
16.02.2009	Anpassungen für OSD V8.0: Alle privilegierten Funktionen von CFS/TAS/JES stehen nun auch unter OSD V8.0 für S/SX und SQ-Server zur Verfügung (--> CFS V17.xx). AB OSD V8.0: Der Parameter KD (Keep Date) bei den Variablen Actions ONXCOPY '...'='...',KD und ONXMOVE :pvs:,KD kann nun auch unter nicht TSOS-Benutzerkennungen angegeben werden. Die neu erzeugte Datei erhält bei KD das Change-Date der ursprünglichen Datei.
12.01.2009	Neue CFS-interne Prozedurvariable &LISTEND2. Im Unterschied zur bereits existierenden Variablen &LISTEND zählt &LISTEND2 auch die verborgenen (mit Action-Code - markierten) Listeneinträge mit.
01.08.2008	AD (Arrange Data): Negative Spaltenangabe (z.B. :-4:4X) ist jetzt möglich. Ein evtl. noch aktives Offset-Kommando (z.B. O-4) wird mit AD zurückgesetzt.
24.06.2008	Neue Anweisung EDTMODE zur Steuerung der Arbeitsmodi des EDT.
24.01.2008	Unterstützung des EDT V17 mit erweiterter Funktionalität: Lange Sätze (> 256 Byte), Unicode Dateien. Der Benutzer kann über das Kommando EDTMODE COMP/UNI steuern, ob bei Vorliegen des EDT V17 der Editor bei dem CFS-Kommando EDT, sowie bei dem Action-Code EDT im kompatiblen (V16.6) oder im neuen Unicode-Modus (V17.0) aufgerufen wird. Der EDT-Modus wird jeweils vor dem ersten Aufruf des Kommandos EDT bzw. des Action-Codes EDT gesetzt und kann danach für die gleiche Aufrufart nicht mehr geändert werden.
28.09.2007	Unter TSOS: Umschalten der aktuellen Sysout-/Syslst-Datei von laufenden Batchtasks.

28.06.2007	Neuer Action-Code DSI: Beim Deallocate Space werden die Dateiattribute Access=Read und Retpd ignoriert.
01.06.2007	Anpassungen für OSD V7.0: Alle privilegierten Funktionen von CFS stehen nun auch unter OSD V7.0 zur Verfügung (--> CFS V16.xx).
19.05.2007	Unterstützung von Unicode-Objekten in CFS.
23.03.2007	Umcodierung von Dateien/Bibliothekselementen gemäß XHCS mit der Variablen Action ONXCONV CCS=name. Ab OSD V7.0 auch zwischen 7-/8-Bit Format und Unicode.
22.12.2006	Erweiterte Unterstützung der ZIP-Archive: Folgende Funktionen sind jetzt auch für Elemente eines ZIP-Archivs verfügbar: Action-Codes C (Copy), R (Rename), E (Erase) Bei E: Keine ERT Funktionalität (Erase with Retain Tempfiles) Action-Code REO, Variable Action ONXREORG, Kommando REORG auf ZIP-Archive
27.09.2006	Neue interne CFS-Variable &TEMP-PREF zeigt die Prefix von Temporärdateien an (S.nnn.) nnn = aktuelle Session-Number. &MONTH/&MONTH2/&MONTH2G: Monat des aktuellen Datums in der Form nn/ January/Januar.
18.09.2006	Mit der Prozeduranweisung *IF %OUT=... können nun auch die SYSOUT-Ausgaben von BS2000-Kommandos in einer Connection geprüft werden.
26.07.2006	Neue Kurzbezeichnung !CFN für benutzereigene Action-Codes. !CFN gibt den Namen der angesprochenen Datei bzw. Job-Variablen. Bei einem Bibliothekselement wird !CFN durch bib-name(t/element/vers) ersetzt.
28.06.2006	Neue User Option NOF
08.05.2006	Neuer Operand in PUBSP-Kommando: SLAVE=Y/N SLAVE=N: Es werden nur Pubsets angezeigt, für die der aktuelle Host der Master ist.
21.03.2006	Connections: In einem Batch-Lauf von CFS kann mit einem neuen Parameter im OC-Kommando der Terminaltyp der zu eröffnenden Connection zu $DIALOG angegeben werden: T9763, T9755, T9750. Nur sinnvoll bei Connections im ES-Modus Beispiel: OC1,,,T9763,ES/LOGON ...
28.11.2005	Neue Routine zum internen Sortieren von Dateienlisten ist im Laufzeitverhalten bei großen Listen wesentlich schneller als die bisherige Routine.
28.11.2005	Kommando RECAT unter TSOS mit Parameter P stellt nun auch verschlüsselte Passworte wieder her.
14.10.2005	CFS-Display/Editor: Beim Display auf Dateien mit einem 8-bit CCS (z.B. EDF041) verwendet CFS automatisch die entsprechende Umsetztabelle zum Anzeigen der Sonderzeichen. Die bisherige benutzerspezifische Umsetztabelle (/FILE CFS.DISP TRTAB,LINK=DISPT) ist damit nicht mehr notwendig.
25.08.2005	Neuer Action-Code FTCF: /CANCEL-FILE-TRANSFER ...,FORCE-CANCEL=YES
26.07.2005	Job-Report-System: Übergabe von Kleinbuchstaben an BS2000-Prozeduren durch PARAM=L'*wert*' bzw. L'*wert*'. L wird in diesem Fall vom Job-Report-System entfernt. *wert* wird ohne die sonst übliche Umsetzung in Großbuchstaben an die BS2000-Prozedur weitergegeben.

14.04.2005 Neue CFS-interne Prozedurvariable &COMP zeigt die Komponente an, in der sich CFS gerade befindet:

SEL	Selektionsmaske
CMD	Kommandofeld
DISP	CFS-Display/Editor
CONx	Connection x

02.02.2005 Anpassungen für OSD V6.0: Alle privilegierten Funktionen von CFS stehen nun auch unter OSD V6.0 zur Verfügung.

15.03.2004 Unterstützung von ZIP-Archiven im BS2000:
Auswahl durch Kommando NP ZIP *zipfile* [, *such*] bzw. ZIP *zipfile* [, *such*] im Feld FILENAME-SELECT. Weitere mögliche Selektionsfelder sind AGE und LASTP. Es sind keine USER-Options vorgesehen.
Aufnahme von Dateien in ZIP-Archiv: ONXZIP *zipfile* [, *params*] oder ONXADD *zipfile*
Selektion von Dateien aus einem ZIP-Archiv: ONXUNZIP ['*str1*'='*str2*'] oder ONXSEL ['*str1*'='*str2*']
Als Action-Codes sind D (Display) und F (Fstat) möglich.

05.11.2003 Kommando PUBSP ,CAT: Es werden für jedes Pubset die Anzahl der belegten und freien Katalogblöcke (4K) angezeigt. Mit dieser Option kann sich der Systemverwalter einen Überblick über evtl. drohende Katalogengpässe bei den einzelnen Pubsets verschaffen.

22.05.2003 Unterstützung des Softwareprodukts SPACEOPT V2.0 von FSC:
Neuer Action-Code SFI (SHOW-FREE-ISAM-SPACE) unter der Kennung TSOS.
Zusätzliche Option ISAM bei User Option FREE (FREE ISAM) unter der Kennung TSOS.

15.05.2003 CFS-Prozedursprache, Anweisung *RUN (URLOUT)... und *RUN (SYSOUT)
Die Anweisungen bieten nützliche Funktionen für unser neues Produkt URLSERVER. Es wird damit die optimale Integration von CFS-Prozeduren in den URLSERVER ermöglicht.

23.09.2002 Anpassung der privilegierten TPR-Module an die neuen SPARC SX-Anlagen (Nachfolger der SR2000). Das Subsystem CAPRI in OSD 5.0 muß dafür gestartet sein.

29.07.2002 Variable Action ONXLIST : Inhaltsverzeichnis der List-Datei im HTML-Format
ONXLIST *datei*, HTML.

10.05.2002 Variable Action ONXMOVE :*catid*:,VOLUME-SET=*xxxx*,S0-MIGRATION=*ALLOWED
Standardmäßig wird bei MOVE auf ein bestimmtes Volume-Set eines PVS das Attribut S0-MIGRATION auf FORBIDDEN gesetzt. Durch den neuen Parameter S0-MIGRATION kann die Eigenschaft nun auf ALLOWED gesetzt werden, wie dies bei der Datei vor der MOVE-Operation in der Regel der Fall ist.

17.04.2002 Variable Action ONXLIST, Inkompatible Optimierung: Bei ONXLIST *datei* mit den Parametern RH (Reduced Header) oder NH (No Header) wird die Ausgabedatei mit BLKSIZE=(STD,16) erstellt. Damit können alle Sätze der Eingabedateien ohne Verkürzung in die Ausgabedatei übernommen werden. Bisher mußte bei 2032 Bytes abgeschnitten werden wegen BLKSIZE=(STD,1).

08.11.2001	Variable Action ONXMOVE: Bisher war es nicht möglich, mit ONXMOVE Pamkey-Dateien auf ein NK-Pubset zu übertragen. Nun versucht CFS nach einem DVS-Fehler 0D80 den Kopiervorgang mit dem Unterprogramm PAMCONV zu bewerkstelligen. Außerdem ist es jetzt mit ONXMOVE möglich, Dateigenerationsgruppen auf ein anderes Pubset zu übertragen. Hierbei werden die Dateigenerationen auf dem Ursprungspubset jedoch nicht gelöscht (gleiche Funktion wie bei ONXCOPY).
24.10.2001	Zusatzparameter LC (Lower Case) bei VMCONS-Connections zur Steuerung einer Linux-VM: OC1 $VMCONS,,*host*,LC
28.09.2001	Variable Action ONXCOPY: Bisher war es nicht möglich, mit ONXCOPY Pamkey-Dateien auf ein NK-Pubset zu kopieren. Nun versucht CFS nach einem DVS-Fehler 0D80 den Kopiervorgang mit dem Unterprogramm PAMCONV zu bewerkstelligen.
22.11.2000	Neue User Option RECSIZE für Dateien. Bestehende User Option RECFORM erweitert in den Selektionsmöglichkeiten und auch auf PLAM-Bibliothekselemente anwendbar.
02.11.2000	Variable Action ONXCONV, CFS-Display Kommando W und S,...=W: Die Dateiattribute (*f-attr*) können außer in der Syntax des FILE-Kommandos nun auch in der Syntax des SDF-Kommandos ADD-FILE-LINK angegeben werden. Mit ? anstelle von *f-attr* wird der SDF-Fragebogen des Kommandos ADD-FILE-LINK angeboten.
05.10.2000	FILENAME-SELECT: Neben FSTAT ... können nun auch Dateien mit dem entsprechenden SDF-Kommando SHFA ... ausgewählt werden. Durch SHFA? wird in den Parameter-Fragebogen des geführten SDF-Dialog verzweigt.
03.10.2000	Systemverwalter (TSOS): Bei eingeschalteter Keep Last Access-Option (Kommando (KLA)) wird bei den Variablen Actions ONXCOPY..,KD, ONXMOVE..,KD und ONXREORG ..,KD neben dem Change Date auch das Last Access Date der beteiligten Dateien wieder hergestellt.
27.07.2000	CFS-Prozeduranweisung */*bs2cmd* kann jetzt auf mehrere Zeilen verteilt werden. Das Fortsetzungszeichen ist '-'. Die Fortsetzungszeile beginnt mit '* '.
29.06.2000	Connections: Kommando RES datei,DELAY=ORIG: Damit werden beim Wiederablaufenlassen eines Dialogs die Eingaben mit der gleichen Zeitverzögerung wie beim Aufzeichnen gesendet.
21.06.2000	Neuer Action-Code FTR: File-Transfer mit Umbenennen der Datei bzw. des Bibliothekselements im Zielsystem.
07.01.2000	Variable Action ONXREN'$*user-id1*.'='$*user-id2*.': Damit können Dateien innerhalb eines Pubsets von einer Kennung in eine andere Kennung umbenannt werden.

Kurzbeschreibung

CFS (Connection & File Services) ist ein Softwaretool für das Betriebssystem BS2000, das durch die unkomplizierte Bedienung und einen großen Leistungsumfang die Effektivität der BS2000-Anwender wesentlich steigert.

Funktionen von CFS (Auszug):

- Auswahl und Bearbeitung von Dateien/Bibliotheken (LMS/FMS/LMR), Jobvariablen und Archive-Directories nach einer einheitlichen Syntax
- Connection-Handler für bis zu 10 Mehrfachdialoge mit einem Datensichtgerät
  - Dialoge aufzeichnen und wieder ablaufen lassen
  - Kommandogedächtnis, Hardcopy des Bildschirms in eine druckaufbereitete Datei
- Editor für Dateien jeder Größe und mit beliebigen Satzlängen, Fcbtyp, Recform, Recsize usw.
- Job-Report System zum Starten von DO-Prozeduren im Enter, maskengesteuertes Prompting der Parameter, Benachrichtigung bei Beendigung der Enter-Aufträge, Anzeige des Sysout-Protokolls
- Find- und Rewrite-Funktion zum Durchsuchen und Ändern beliebig vieler Dateien/Bibliotheks-elemente mit einem Kommando
- Datei-Informations- und Dokumentationssystem (Beschreibung zu Dateien hinterlegen)
- Maskengesteuerte Unterstützung aller File-Transfer Systeme
- logisches Vergleichen von Dateien/Bibliotheken/Benutzerkennungen/Public-Volume-Sets
- Konvertieren von Dateien ohne Einschränkung für alle Dateiformate
- Reorganisieren von Bibliotheken, ZIP-Archiven und ISAM-Dateien
- In Archive-Directories gesicherte Dateien auswählen und bei Bedarf unter dem originalen oder einem geänderten Namen einspielen
- benutzerspezifische Terminverwaltung mit Kalenderfunktion
- spezielle Funktionen für den Systemverwalter:
  - Auswahl aller geöffneten Dateien im System zusammen mit Open-Modus, TSN, ISAM-Pool
  - Inhalt von geöffneten SAM-/PAM-Dateien anzeigen (z.B. SYSOUT-Protokoll/CONSLOG-Datei)
  - Dateien auf anderen PVS kopieren/moven unter Beibehaltung des Change Date
  - Auswahl aller Dateien mit einem bestimmten Passwort
  - Speicherbelegung der Platten aller Public-Volume-Sets bzw. aller Privatplatten anzeigen
- alle Funktionen auch in Prozeduren verfügbar
- Hilfe-Informationen an jeder Stelle im Programm

CFS besitzt Schnittstellen zu folgenden Produkten: EDT, TOM-TI, EDOR, FMS, LMS, LMR, ARCHIVE, FLAM, PERCON, ESR, SORT, openFT, RDAC, $DIALOG, $CONSOLE, DCAM, UTM, SAP.

Die Installation von CFS erfolgt einfach durch Einspielen eines Magnetbandes unter einer beliebigen Benutzerkennung. Von Seiten des Systemverwalters sind keine Änderungen im BS2000 vorzunehmen. Beim Wechsel auf eine höhere BS2000-Version sind keinerlei Anpassungen notwendig.
CFS ist ablauffähig unter BS2000/OSD V1.0 bis OSD V8.0.

Anwendungsbeispiele

Aufruf von CFS

/EXEC $CFS   bzw. /EXEC $CFS.CFS

Beendigung von CFS

Eingabe * oder END im Kommandofeld oder wiederholte Betätigung der K1-Taste, K2-Taste.

Verlassen von Menues

Mit der K1-Taste kann der Benutzer jede CFS-Maske verlassen und in das hierarchisch darüber liegende Menue zurückkehren.

Selektion von Dateien und Aufnahme in eine Bibliothek

Im Kommandofeld der CFS-Dateienliste wird ONXADD *bibliothek* eingegeben. Die in die Bibliothek aufzunehmenden Dateien werden ausgewählt durch Eintragen des Buchstabens X (Action-Code) in der rechten Eingabespalte der Dateienliste. Nach Eingabe des Kommandos A wird die Aufnahme aller vorgemerkten Dateien in die Bibliothek veranlaßt. Der Action-Code XE bewirkt, daß die Dateien nach erfolgreicher Aufnahme in die Bibliothek gelöscht werden.

Ändern der Datei-Attribute

Durch AR in der Action-Spalte einer Datei wird diese auf Access=Read gesetzt. AW bewirkt eine Umkatalogisierung nach Access=Write, SY: Share=Yes, SN: Share=No. Mit dem Action-Code F (Fstat) kann die vorgenommene Änderung der Dateiattribute verifiziert werden.
Durch den Buchstaben R (Rename) wird der Name der betreffenden Datei überschreibbar und kann auf einfache Weise geändert werden. Durch den Buchstaben C (Copy) wird der Name ebenfalls überschreibbar. Es kann der Name einer zu erzeugenden Kopie der markierten Datei angegeben werden.

Bearbeiten von Datenobjekten im EDT

Durch Eingabe von EDT in der Action-Spalte einer Datei wird diese in der Ebene 0 des EDT bereitgestellt. Es können auch mehrere Dateien gleichzeitig in verschiedenen Ebenen des EDT bearbeitet werden durch Markieren mit EDT0, EDT1, EDT2, usw. Nach Beendigung des Editiervorgangs ist die K1-Taste zu drücken: Es erscheint UPD0, UPD1, UPD2, usw. in den Action-Spalten der zuvor markierten Dateien. Durch unverändertes Absenden dieser Maske, bzw. durch Überschreiben mit Blanks wird das gezielte Zurückschreiben der einzelnen Dateien gesteuert.

Auswahl von Elementen einer Bibliothek

Selektionsmaske, Feld FILENAME-SELECT: Eingabe von LMS *bibliothek* / FMS *bibliothek* zeigt alle Elemente der betreffenden Bibliothek an. Die weitere Bearbeitung der Bibliothekselemente (z.B. mit EDT) wird in der auch für Dateien gültigen Weise vollzogen (siehe oben).

Auswahl von Jobvariablen

Selektionsmaske, Feld FCBTYPE: Eingabe von J (Jobvariables).

Anzeigen und Ändern des Inhalts von Dateien/Bibliothekselementen/Jobvariablen

Der Buchstabe D im Action-Feld bewirkt, daß die so markierte Datei/Jobvariable, bzw. das markierte Bibliothekselement in einem CFS-eigenen Editor angezeigt wird. Positionieren innerhalb der Datensätze (+*n*/-*n*/++/--/>*n*/<*n*/>>/<<) und das Setzen verschiedener Anzeigemodi (HEX/EL) erfolgt über das Kommandofeld (COMMAND :). Das Datenobjekt wird zum Ändern freigegeben, sobald das Kommando M (Modify) eingegeben wurde. Das Verlassen des Display/des Editors geschieht durch Drücken der K1-Taste.

Begrenzten Ausschnitt eines Datenobjekts auf Drucker ausgeben

Durch den Action-Code D wird der Display-Modus für die Datei/Jobvariable/das Bibliothekselement aktiviert. Über Positionier-, bzw. Suche-Kommandos (S,'...') werden die interessierenden Daten am Bildschirm dargestellt, wobei durch EL/HEX der Anzeigemodus verändert werden kann. Durch Drücken der F3-Taste wird der gerade angezeigte Bildschirmausschnitt in einer druckaufbereiteten Datei festgehalten. Diese kann anschließend mit dem Kommando NP CFS.HARDC am Bildschirm angezeigt und mit dem Action-Code PS mit Space=E ausgedruckt werden.

Mehrfachdialoge unter CFS

Eintragen von OC1 im Kommandofeld der zweiten CFS-Maske. Es erscheint daraufhin die BS2000-Meldung: "PLEASE LOGON ... ". Durch Eingabe eines korrekten LOGON-Kommandos wird ein Dialogprozeß mit einer eigenen TSN erzeugt. Durch Drücken der K2-Taste wird ein Prompting ausgegeben, dessen Beantwortung durch den Buchstaben R (Return) den Benutzer wieder in die Basisanwendung von CFS führt. Es kann nun durch OC2 eine weitere Connection eröffnet, bzw. durch das Kommando CC1 (Continue in Conn. 1) in die mit OC1 eröffnete Verbindung wieder zurückverzweigt werden.

Falls der Benutzer vor allem an der Möglichkeit der Mehrfachdialoge interessiert ist, so kann der Schritt der Dateiauswahl übersprungen werden durch Eingabe von NO im Feld FILENAME-SELECT.

Dialoge mitschneiden und wieder ablaufen lassen

Um den Mitschnitt aller Dialogeingaben zu veranlassen, ist in einer Connection das Kommando LOG *datei* einzugeben. Alle nachfolgenden Eingaben werden sodann in der angegebenen Datei mitgeschnitten. Eine Beendigung des Mitschnitts wird durch das Kommando NLOG erreicht. Um den Dialog in der gleichen oder in einer späteren Sitzung wieder ablaufen zu lassen, ist das Kommando RES *datei* anzugeben. Um den Dialog schrittweise mit Benutzerbestätigung ablaufen zu lassen, ist das Kommando RES *datei*,S einzugeben.

DO-Prozeduren mit Parametern im Enter starten (Job-Report)

Durch das Job-Report System von CFS können DO-Prozeduren in Enter-Aufträgen zum Ablauf gebracht werden. Im Feld COMMAND in der zweiten CFS-Maske ist hierzu das Kommando E *do-proc* einzugeben. Enthält die PROC-Anweisung der DO-Prozedur Stellungs- oder Schlüsselwortparameter, so werden diese in Form einer Maske angezeigt und können gegebenenfalls vervollständigt werden. Sobald der von CFS gestartete Enter-Auftrag beendet ist, erhält der Benutzer davon Nachricht und

kann das SYSOUT-Listing des Enter-Auftrags betrachten. Befindet sich der Benutzer zum Zeitpunkt der Beendigung des Enter-Prozeßes nicht mehr im Programm CFS, so wird er beim nächsten Aufruf automatisch über die Beendigung dieses Enter-Auftrags informiert.

File-Transfer Aufträge starten und überwachen

Durch Eingabe des Kommandos FT im Feld COMMAND wird eine Maske zum Übertragen von Dateien mit openFT aufgerufen. Hier werden alle wesentlichen Angaben für den File-Transfer Auftrag eingetragen. Über ? kann eine detaillierte Auskunft zu den einzelnen Maskenfeldern angefordert werden. Durch ?FT, eingegeben im Kommandofeld von CFS, wird eine ausführliche Einführung in das gesamte FT-System geboten.

Falls in der FT-Maske im Feld PROTOCOL-LISTING : YES eingegeben wurde, so wird das Job-Report System (siehe letzter Abschnitt) bei Beendigung des FT-Auftrags aktiviert und das FT-Protokoll wird am Bildschirm angezeigt.

Durch Eingabe von NSTAT im Feld FILENAME-SELECT in der ersten CFS-Maske wird eine Übersicht aller Dateien angezeigt, für die ein File-Transfer gestartet wurde. Durch FTS, eingegeben in der Action-Spalte kann der Übertragungsstatus für einzelne Dateien noch detaillierter angefordert werden. Durch den Action-Code FTC wird der Transfer-Auftrag für die angekreuzte Datei zurückgenommen (gecancelt).

Allgemeine Vereinbarungen (Metazeichen)

In dem vorliegenden Benutzerhandbuch werden bestimmte Zeichen (sogenannte Metazeichen) zur Darstellung der verschiedenen Eingabemöglichkeiten verwendet. Hierbei werden die in der nachfolgenden Tabelle aufgeführten Vereinbarungen getroffen:

Formale Darstellung	Erläuterung	Beispiel
GROSS	Großbuchstaben bezeichnen Konstanten, die in dieser Form vom Benutzer eingegeben werden müssen.	SO AGE Einzugeben ist: `SO AGE`

kursiv	kursive Kleinbuchstaben bezeichnen Variablen, die bei der Eingabe vom Benutzer durch aktuelle Werte ersetzt werden müssen, d.h. ihr Inhalt wird von Fall zu Fall verschieden sein.	ONXADD *bibliothek*
		
		Einzugeben ist:
		`ONXADD LMS.PROC`
		`ONXADD $TSOS.MACROLIB`
		usw.

[.....]	Eckige Klammern schließen Wahleingaben ein. Diese können angegeben oder weggelassen werden. Steht bei Wahleingaben ein Komma innerhalb der Klammer, so wird es nur bei Verwendung dieser Eingabe verlangt.
	Runde Klammern (..) müssen stets so eingegeben werden wie beschrieben.
		LOG [*bibl*](*elem*)
		
		Einzugeben ist:
		`LOG XYZ.BIBL(TEST)`
		`LOG (TEST)`

|	Der senkrechte Strich trennt alternativ zu verwendende Eingaben. Es ist dabei von Bedeutung, ob die senkrechten Striche von eckigen oder geschweiften Klammern eingeschlossen werden.	LOG [*datei* | *bibl*(*elem*)]
		
		Einzugeben ist:
		`LOG CFS.LOG`
		`LOG CFS.LIB(TEST)`
		`LOG`

{ ..... }	Geschweifte Klammern schließen Alternativen ein, d.h. aus den eingeschlossenen Größen muß eine Angabe ausgewählt werden.	LOG {*datei* | *bibl*(*elem*) }
		
		Einzugeben ist:
		`LOG CFS.LOG`
		`LOG CFS.USERLIB(TEST)`

Formale Darstellung	Erläuterung	Beispiel

AAAAAA (unterstrichen)	Die Unterstreichung hebt den Standardwert (Voreinstellung) hervor. Dies ist der Wert, der von CFS eingesetzt wird, wenn der Benutzer keine Angabe macht.	SO [AGE | NAME | SIZE]
		Einzugeben ist:
		`SO AGE`
		`SO NAME`
		`SO SIZE`
		`SO` (hat die gleiche Wirkung wie `SO NAME`)

⊔	Dieses Symbol kennzeichnet ein Leerzeichen (X'40').	DO⊔ Einzugeben ist: `DO` + Leerzeichen
◀	Dieses Symbol kennzeichnet die Endemarke auf der Tastatur. Bei einigen Tastaturen ist die Endemarke auch mit EM	--◀ Einzugeben ist: `--` + Endemarke
ENTER **DUE1** Taste	Mit ENTER wird die normale Datenübertragungs-Taste bezeichnet. Bei älteren Terminals ist die entsprechende Taste mit DUE1 beschriftet.	--◀ **ENTER** Einzugeben ist: `--` + Endemarke + ENTER-
Shift ENTER **DUE2**	Shift Enter steht für die Datenübertragungsfunktion DUE2. Shift bezeichnet die Umschalttaste für Großbuchstaben. Bei älteren Terminals gibt es eine eigene Taste mit der Beschriftung DUE2.	**-- Shift ENTER** Einzugeben ist: `--` + SHIFT (Umschalten auf Großbuchstaben) + ENTER
Shift Reset Bild **LSP**	Shift Reset Bild steht für die Terminalfunktion "Löschen Bildschirmspeicher". Shift bezeichnet die Umschalttaste für Großbuchstaben. Bei älteren Terminals gibt es eine eigene Taste mit der Beschriftung LSP (Löschen Bildspeicher).	
K1 **K2** **K3**	Diese Symbole kennzeichnen die gleichnamigen Datenübertragungstasten für die Kurznachrichten K1, K2, K3, usw.	
F1 **F2**	Diese Symbole kennzeichnen die gleichnamigen Datenüber-	



tragungstasten F1, F2, F3, usw.

Fachwörter

Action-Code

Ein Action-Code ist eine bis zu fünfstellige alphanumerische Zeichenfolge, die in der Action-Spalte jedes Eintrags der Dateienliste angegeben werden kann. Action-Codes stehen für einfache Kommandos und haben keine Parameter. Will man bestimmte Verarbeitungsoptionen nur für eine oder wenige in der Liste aufgeführte Datenobjekte durchführen, so vereinfacht sich das Handling durch Verwendung der Action-Codes.

Connection

Eine Connection ist ein eigenständiger Dialog mit $DIALOG oder einer DCAM-/UTM-Anwendung, den der Benutzer jederzeit eröffnen und wieder beenden kann. Es sind bis zu 10 Connections pro geladenem CFS möglich. Das Wechseln zwischen den einzelnen Connections geschieht durch Drücken der K2-Taste und nachfolgender Eingabe der gewünschten Connection-Nummer.

Dateienliste

Nach der Eingabe der Selektionsbedingungen erscheint am Bildschirm eine Maske, in der die ausgewählten Datenobjekte (Dateien/Jobvariablen/Bibliothekselemente) in einer übersichtlichen Weise formatiert dargestellt werden.
Zur Eingabe im Kommandofeld der Dateienliste stehen neben allen BS2000-Kommandos eine Vielzahl zusätzlicher CFS-Kommandos zur Dateibearbeitung zur Verfügung. Außerdem bietet die Dateienliste für jedes angezeigte Datenobjekt eine Action-Spalte zum Eintragen von Action-Codes.

Hardcopy

Die Ein-/Ausgaben am Bildschirm werden von CFS zeilen- und spaltengerecht in einer druckaufbereiteten Datei protokolliert. Der Hardcopy-Modus kann in allen Masken von CFS, sowie in jeder von CFS eröffneten Connection eingeschaltet werden.

Help-System

Das Help-System ist eine Einrichtung in CFS, die es dem Benutzer erlaubt, sich auch am Bildschirm die im CFS-Benutzerhandbuch enthaltenen Informationen auf einfache und strukturierte Weise zu vergegenwärtigen. Darüber hinaus kann zu jedem Kommando, sowie zu jedem Eingabefeld einer Maske gezielt die entsprechende Hilfe-Information abgefragt werden.

Job-Report

Das Job-Report System ermöglicht das Starten von DO-Prozeduren im Enter-Modus. Für die DO-Prozedur notwendige Parameter werden von CFS erkannt und ggf. in einer eigenen Maske (Parameter-Maske) gepromptet. Nach Beendigung des Enter-Auftrags wird der Benutzer durch Ausgabe des formatierten SYSOUT-Protokolls automatisch über den Ablauf des Auftrags informiert.

Kommandogedächtnis

Der Benutzer hat die Möglichkeit, sich früher getätigte Eingaben am Bildschirm wieder anzeigen zu lassen und diese unverändert oder in abgeänderter Form erneut zur Ausführung zu bringen. Die Einrichtung des Kommandogedächtnisses steht zur Verfügung im Feld FILENAME-SELECT der Selektionsmaske, im Kommandofeld der Dateienliste, sowie bei allen Line-Mode Eingaben in beliebigen Connections. Das Kommandogedächtnis wird in allen CFS-Masken durch die Tasten Shift ENTER

(DUE2) aktiviert. In Connections wird das Gedächtnis durch Eingabe des Zeichens '-' und Shift ENTER aktiviert.

Logging

Logging ist ein Zusatzservice von CFS in Connections. Bei eingeschaltetem Logging-Modus werden alle in dieser Connection getätigten Eingaben im physikalischen Format, d.h. mit Nachrichtenkopf und allen Feldattributen in einer ISAM-Datei protokolliert (mitgeschnitten). Für besondere Zwecke können neben Eingaben auch alle Bildschirmausgaben mitgeschnitten werden. Zum gleichen Thema siehe auch "Restore".

Restore

Restore ist ein Zusatzservice von CFS in Connections. Mit Aktivierung des Restore wird das Eingabemedium für diese Connection von der Tastatur auf eine im Logging-Modus (siehe oben) erstellte Datei gelegt. Damit ist es möglich, einmal erfaßte Dialoge beliebig oft und ohne Eingaben durch den Benutzer wieder ablaufen zu lassen. Im Restore-Modus sind verschiedene Varianten möglich:

a) automatisches Ablaufenlassen eines Dialogs ohne Benutzereingabe (Batch-Modus).
b) Ablaufenlassen des Dialogs, wobei der Benutzer jede aus der LOG-Datei gelesene Eingabe durch Drücken der Enter-Taste bestätigen muß (Step-Modus).
c) automatisches Ablaufenlassen eines Dialogs, wobei die aktuell empfangenen Ausgabenachrichten mit den mitgeschnittenen Ausgaben verglichen werden. Bei Ungleichheit wird der Benutzer informiert (Compare-Modus).
d) Simulation des mitgeschnittenen Dialogs am Bildschirm. Alle Ein- und Ausgaben werden nur am Bildschirm dargestellt und nicht an den Kommunikationspartner gesendet bzw. von diesem empfangen (Phantom-Modus).

Selektion

Nach dem Programmaufruf wird dem Benutzer die leere Selektionsmaske angeboten. Hier kann er sich überlegen, mit welchen Datenobjekten er in nächster Zeit arbeiten möchte. Durch entsprechende Eingaben in der Selektionsmaske können diese Datenobjekte ausgewählt werden. Z.B. Dateien mit bestimmten Eigenschaften (Suchbegriffe im Namen, bestimmtes Änderungsdatum und/oder Fcbtyp, usw.), Elemente einer Bibliothek mit bestimmten Eigenschaften oder Jobvariablen.

Falls **keine Selektion** von Datenobjekten erfolgen soll, z.B. weil der Benutzer lediglich Connections eröffnen möchte, so ist im ersten Feld der Selektionsmaske "NO" einzutragen.

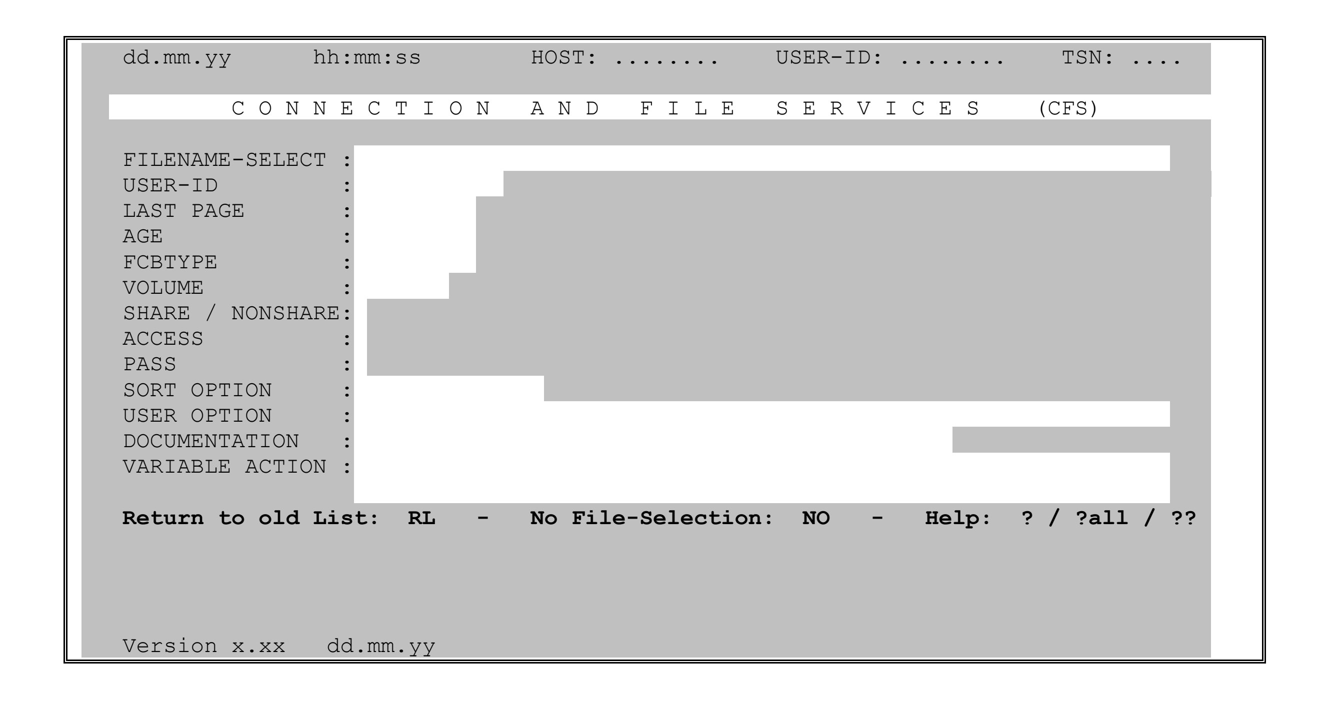
Startup

Eine nach dem Aufruf von CFS automatisch ausgeführte Folge von Kommandos. Die auszuführenden Kommandos sind in einer Datei unter der Benutzerkennung des Aufrufers abgelegt. Der Name der Startup-Datei muß in einer festgelegten Art und Weise gebildet sein (CFS. STARTUP. ...).

Variable Action

Eine Variable Action ist eine komplexe Verarbeitungsinstruktion, die auf eine gezielt ausgewählte Gruppe von Datenobjekten angewendet werden kann. Die durch Variable Actions bezeichneten Verarbeitungsinstruktionen können durch Parameter modifiziert werden. Damit ist es möglich, eine zuvor beliebig zusammengestellte Menge von Dateien/Jobvariablen/Bibliothekselementen mit einem einzigen Kommando einheitlich zu bearbeiten.

Selektionsmaske



```
dd.mm.yy      hh:mm:ss        HOST: ........    USER-ID: ........    TSN: ....

        C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -   No File-Selection:  NO   -   Help:  ? / ?all / ??



Version x.xx   dd.mm.yy
```

Die Selektionsmaske ist das "Einstiegsbild" von CFS. In dieser Maske werden der Typ und weitere spezielle Eigenschaften von Datenobjekten festgelegt, die für den Benutzer nachfolgend besonders interessant sind.

Im Feld "SORT OPTION" kann das Merkmal angegeben werden, nach dem die Dateien-liste, das Ergebnis der Selektion, zu sortieren ist (z.B. aufsteigend nach Alter der Datei-en, absteigend nach Größe oder unsortiert, d.h. in der Reihenfolge wie die Dateien im Katalog abgelegt sind).

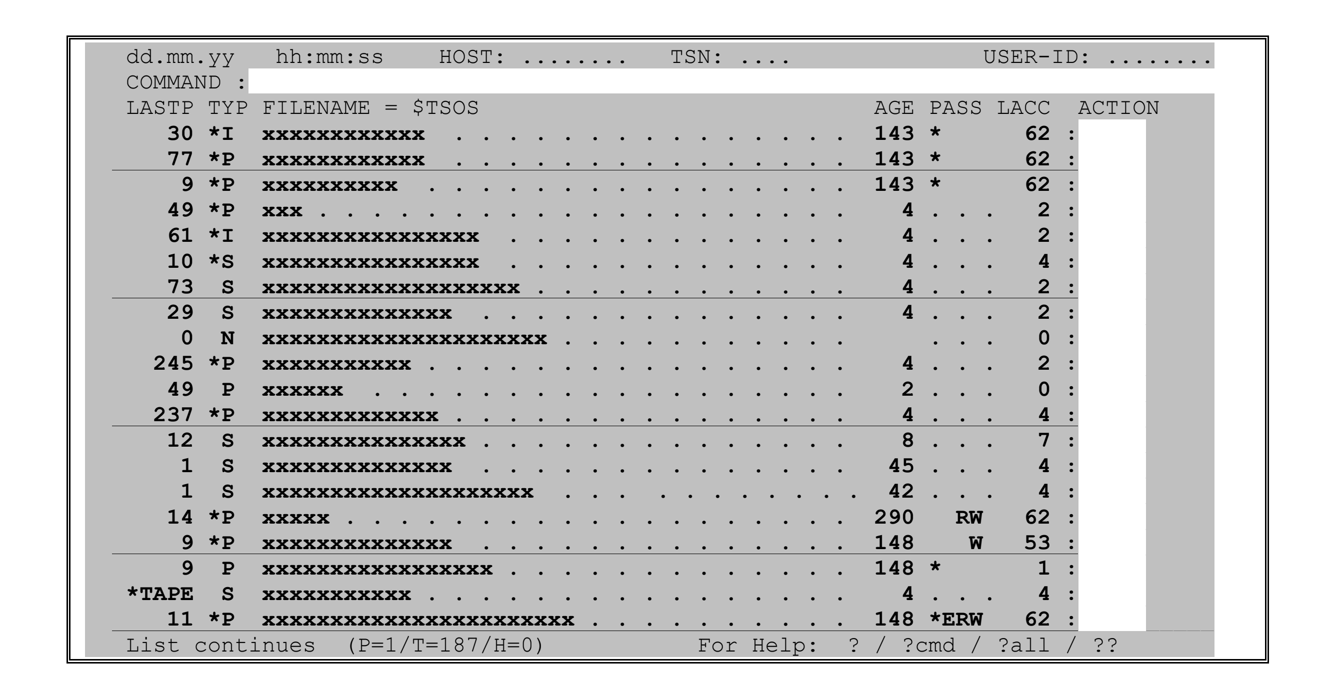
Im Feld "USER OPTION" kann ein zusätzliches, vom Benutzer frei definierbares Merkmal angegeben werden, das in der Dateienliste angezeigt wird und nach dem die Datenob-jekte gegebenenfalls ausgewählt werden.

Im Feld "DOCUMENTATION" kann eine Datei angegeben werden, in die die selektierte und am Bildschirm angezeigte Liste zusätzlich geschrieben wird.

Im letzten Feld der Selektionsmaske kann eine sog. "Variable Action" definiert werden. Diese geben dem Benutzer die Möglichkeit, komplexere Verarbeitungen auf eine gezielt ausgewählte und in der Regel größere Menge von Dateien, Jobvariablen oder Biblio-thekselemente anzuwenden.

Falls der Benutzer keine Datenobjekte selektieren möchte, so kann er durch Eingabe von NO im Feld FILENAME-SELECT sofort in die zweite CFS-Maske verzweigen.

Dateienliste



```
dd.mm.yy   hh:mm:ss    HOST: ........   TSN: ....              USER-ID: ........
COMMAND :
LASTP TYP FILENAME = $TSOS                                 AGE  PASS LACC   ACTION
   30 *I  xxxxxxxxxxxx  . . . . . . . . . . . . . . . .    143 *       62 :
   77 *P  xxxxxxxxxxxx  . . . . . . . . . . . . . . . .    143 *       62 :
    9 *P  xxxxxxxxxx  . . . . . . . . . . . . . . . . .    143 *       62 :
   49 *P  xxx . . . . . . . . . . . . . . . . . . . . .      4 . . .    2 :
   61 *I  xxxxxxxxxxxxxxxx  . . . . . . . . . . . . . .      4 . . .    2 :
   10 *S  xxxxxxxxxxxxxxxx  . . . . . . . . . . . . . .      4 . . .    4 :
   73  S  xxxxxxxxxxxxxxxxxxx . . . . . . . . . . . . .      4 . . .    2 :
   29  S  xxxxxxxxxxxxxx  . . . . . . . . . . . . . . .      4 . . .    2 :
    0  N  xxxxxxxxxxxxxxxxxxxxx . . . . . . . . . . . .        . . .    0 :
  245 *P  xxxxxxxxxxx . . . . . . . . . . . . . . . . .      4 . . .    2 :
   49  P  xxxxxx  . . . . . . . . . . . . . . . . . . .      2 . . .    0 :
  237 *P  xxxxxxxxxxxxx . . . . . . . . . . . . . . . .      4 . . .    4 :
   12  S  xxxxxxxxxxxxxxx . . . . . . . . . . . . . . .      8 . . .    7 :
    1  S  xxxxxxxxxxxxxx  . . . . . . . . . . . . . . .     45 . . .    4 :
    1  S  xxxxxxxxxxxxxxxxxxxx  . . .   . . . . . . . .     42 . . .    4 :
   14 *P  xxxxx . . . . . . . . . . . . . . . . . . . .    290   RW    62 :
    9 *P  xxxxxxxxxxxxxx  . . . . . . . . . . . . . . .    148    W    53 :
    9  P  xxxxxxxxxxxxxxxxx . . . . . . . . . . . . . .    148 *        1 :
*TAPE  S  xxxxxxxxxxx . . . . . . . . . . . . . . . . .      4 . . .    4 :
   11 *P  xxxxxxxxxxxxxxxxxxxxxxx . . . . . . . . . . .    148 *ERW    62 :
List continues  (P=1/T=187/H=0)           For Help:  ? / ?cmd / ?all / ??
```

In dieser Maske wird dem Benutzer das Ergebnis der Selektion präsentiert. Es werden alle Dateien/ Jobvariablen/Bibliothekselemente aufgeführt, die die angegebenen Auswahlbedingungen erfüllen.

Die Maske enthält im oberen Teil ein Kommandofeld. Hier können BS2000-Kommandos und spezielle CFS-Kommandos eingegeben werden.

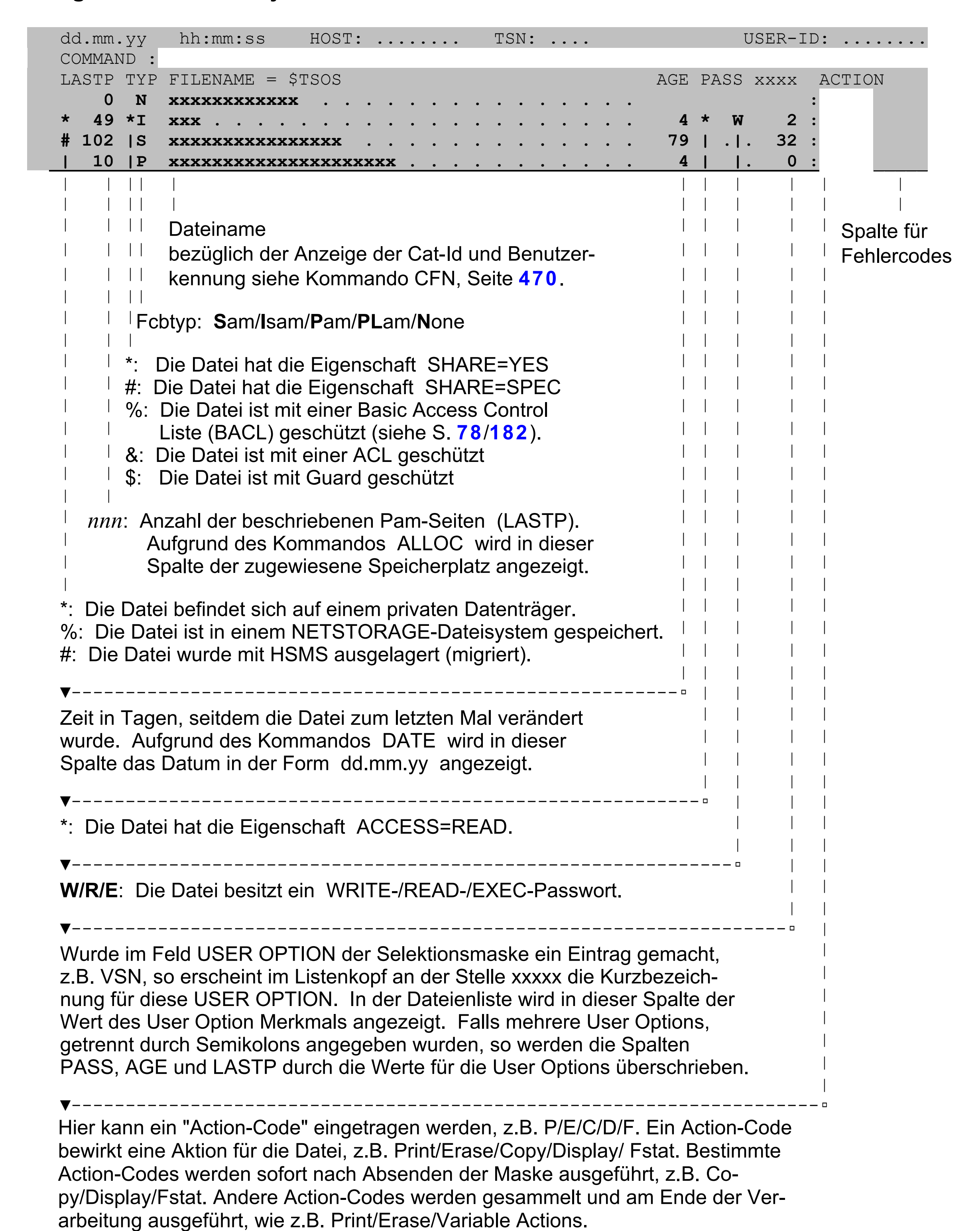
Für jedes in der Dateienliste aufgeführte Datenobjekt (Datei/Jobvariable/Bibliothekselement) existiert rechts ein Action-Feld. Über einen mnemotechnischen Kürzel (Action-Code) können damit bestimmte Aktionen auf das Datenobjekt angewendet werden.

Die Dateienliste ist standardmäßig nach den Dateinamen aufsteigend sortiert. Es können auch andere Sortierfelder vorgegeben werden. Siehe hierzu Kommando SO auf Seite 482.

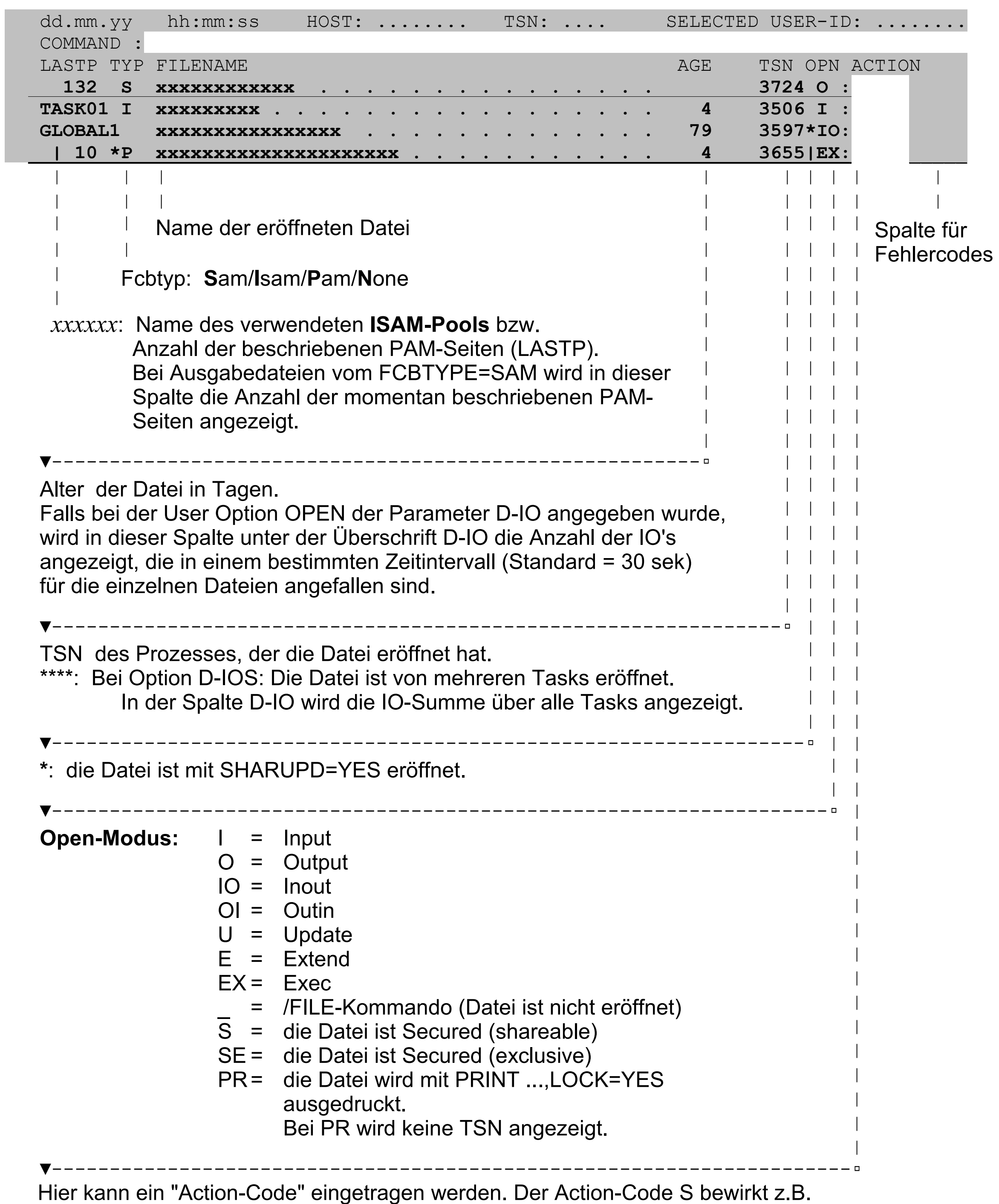
In der letzten Zeile werden allgemeine Angaben zur Dateienliste angezeigt.

P=*ppp*	Aktuelle Position der ersten am Bildschirm dargestellten Zeile innerhalb der Liste.
T=*nnn*	Gesamtanzahl der in der Dateienliste vorhandenen Einträge (Total).
H=*qqq*	Anzahl der durch Action-Code - bzw. E unsichtbar gemachten/gelöschten Einträge.

Ausgewählte Datenobjekte: Dateien

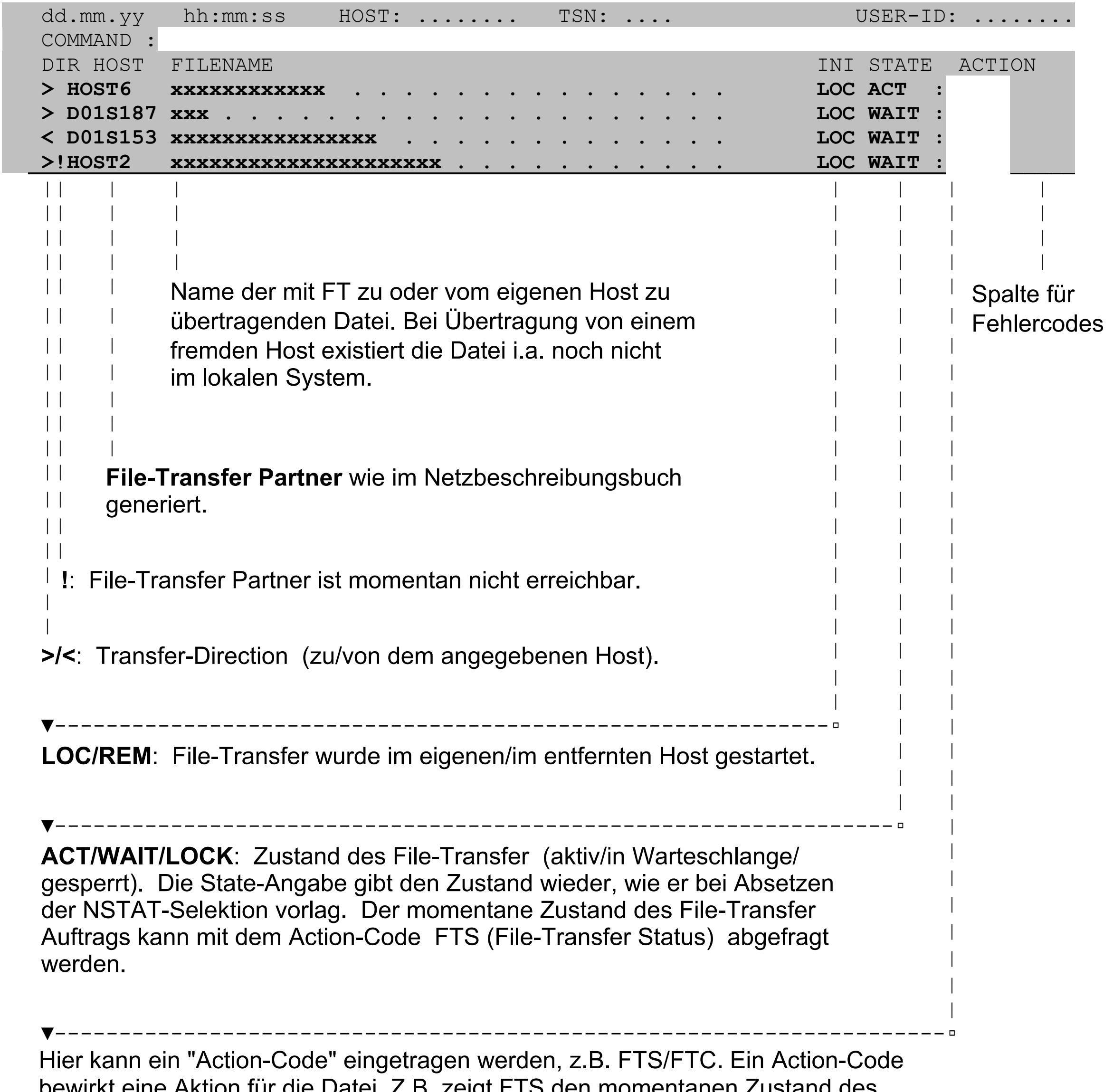


Ausgewählte Datenobjekte: geöffnete Dateien (User Option OPEN)



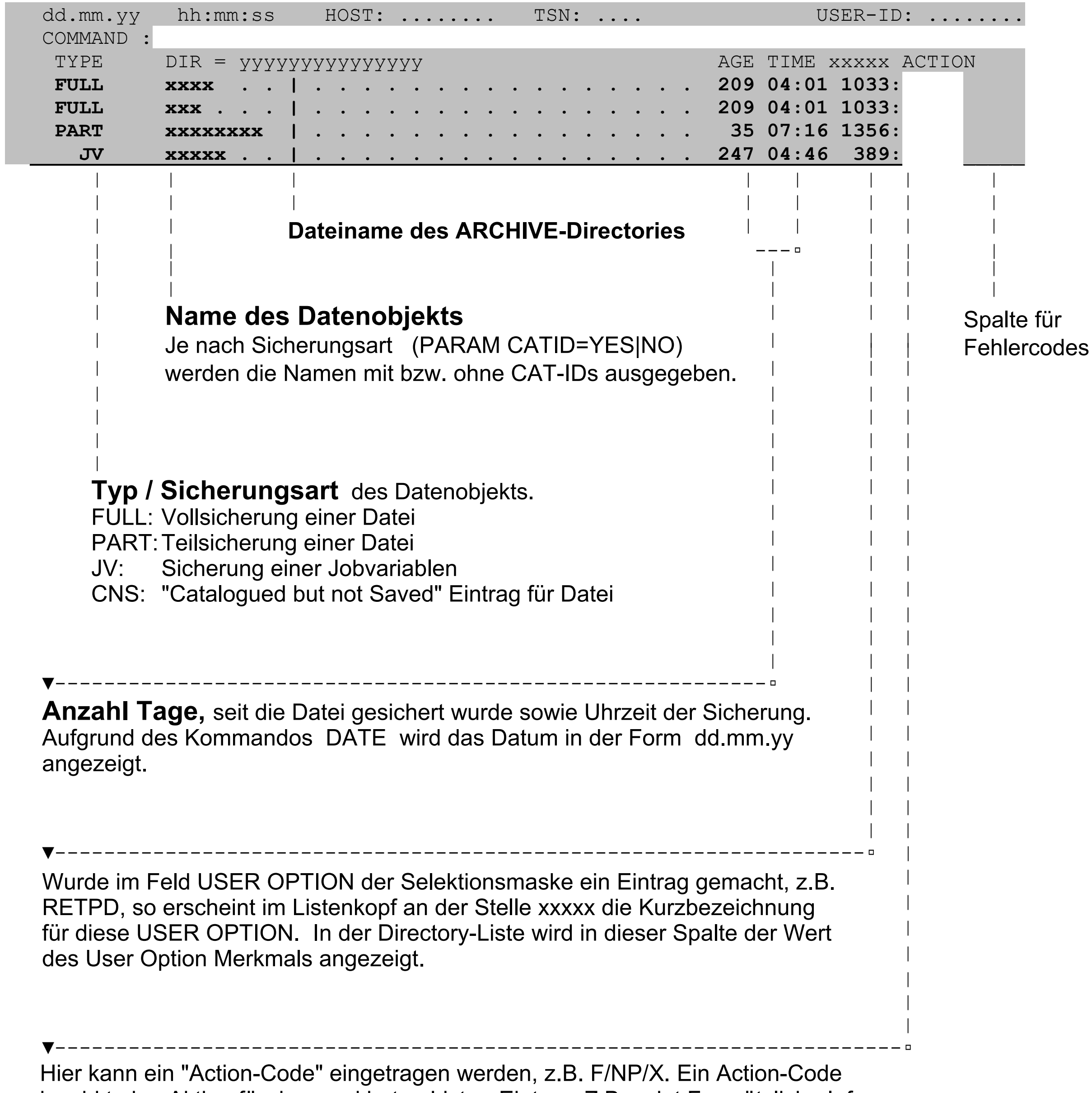
Hier kann ein "Action-Code" eingetragen werden. Der Action-Code S bewirkt z.B. die Ausgabe der Status-Informationen zur angezeigten TSN. Der Action-Code D (Display) kann auch auf SAM-Dateien angewendet werden, die im Output-Modus eröffnet sind, wie z.B. Ablaufprotokolle.

Ausgewählte Datenobjekte: Mit FT übertragene Dateien (NSTAT)



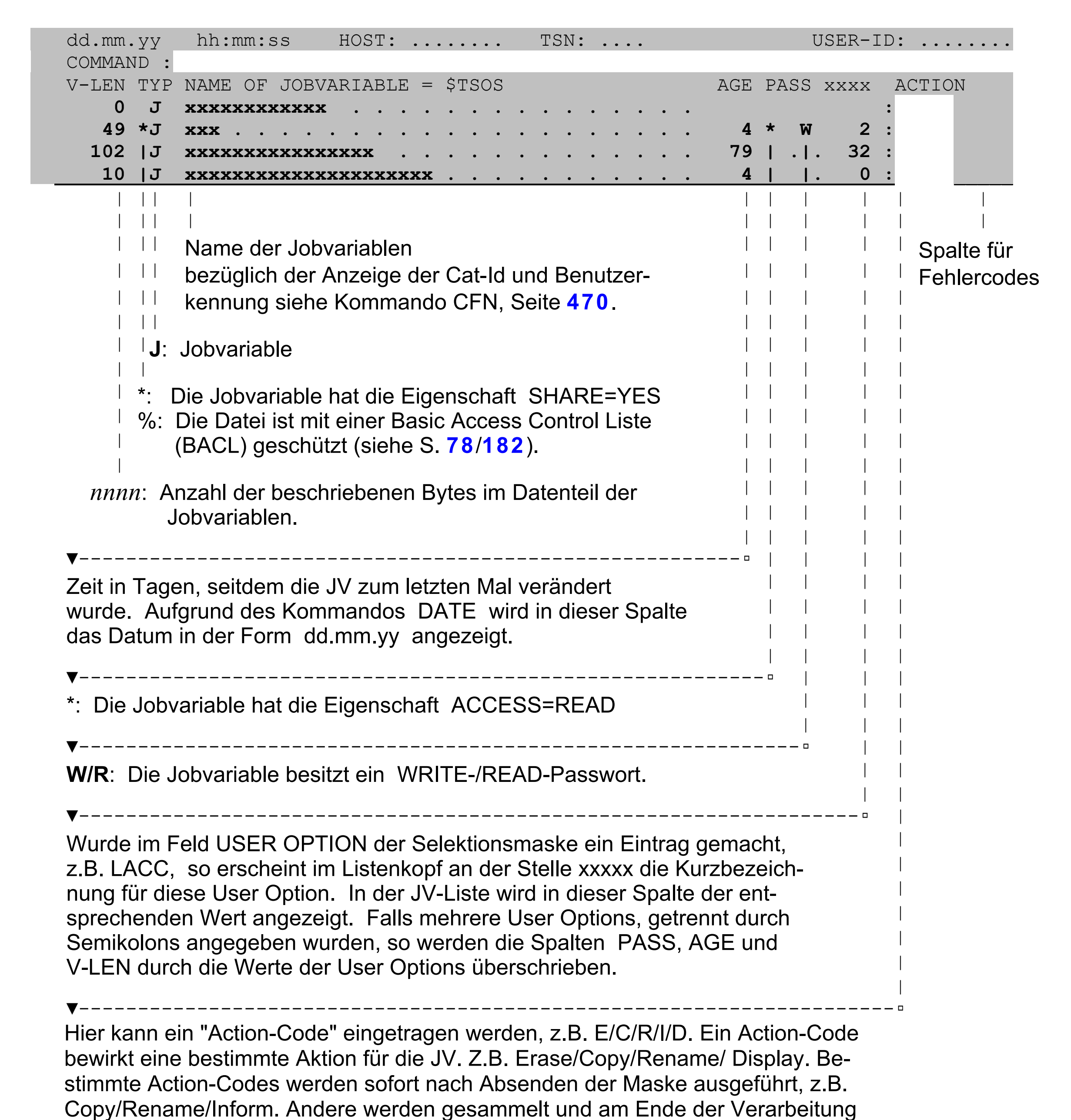
Hier kann ein "Action-Code" eingetragen werden, z.B. FTS/FTC. Ein Action-Code bewirkt eine Aktion für die Datei. Z.B. zeigt FTS den momentanen Zustand des File-Transfer Auftrags an. Mit dem Action-Code FTC kann der File-Transfer Auftrag annulliert (gecancelt) werden. Für weitere Erläuterungen der Action-Codes wird auf das Kapitel 6 "Action-Codes" verwiesen.

Ausgewählte Datenobjekte: Archive-Directory Einträge (DIR)

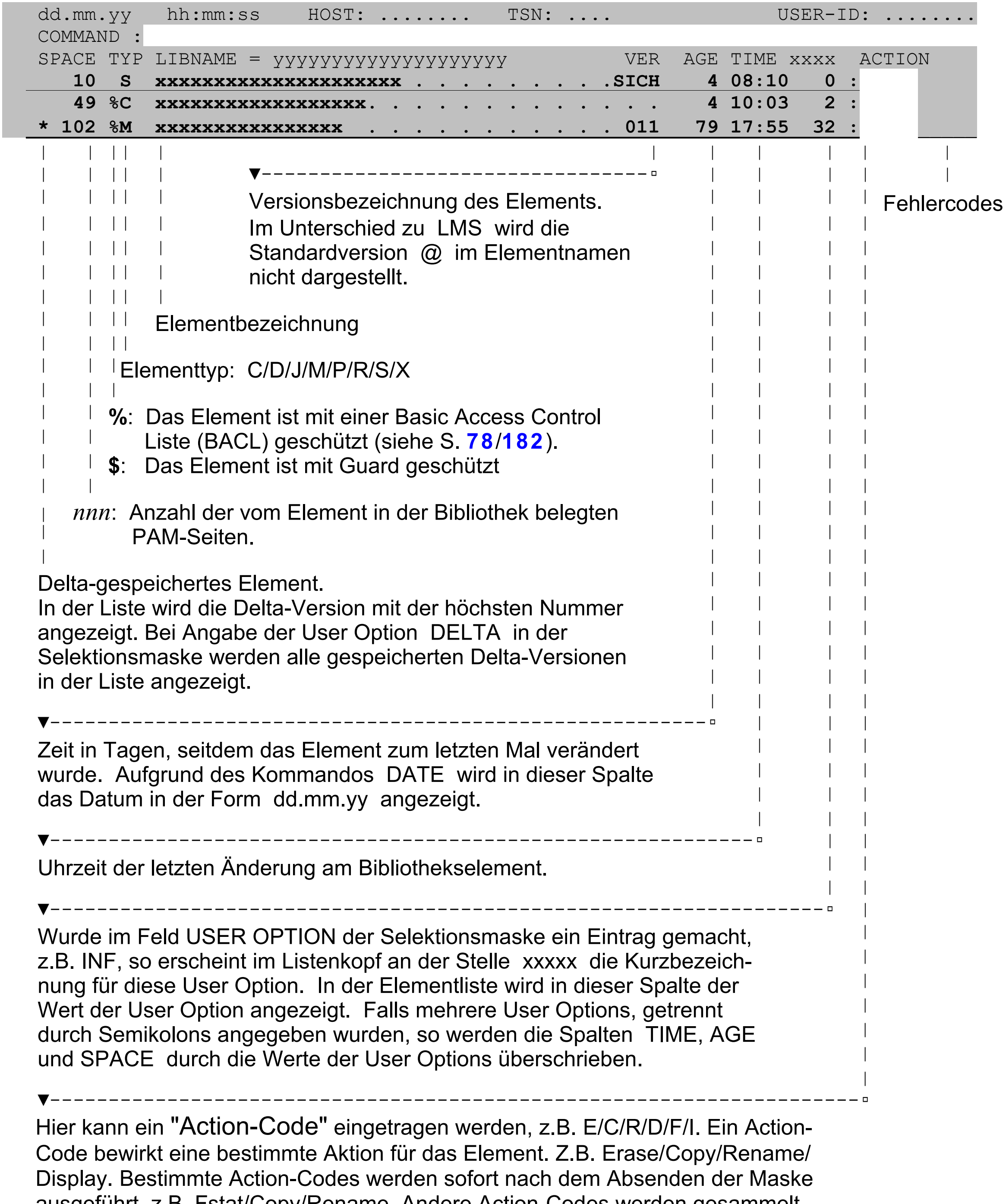


Hier kann ein "Action-Code" eingetragen werden, z.B. F/NP/X. Ein Action-Code bewirkt eine Aktion für den markierten Listen-Eintrag. Z.B. zeigt F zusätzliche Informationen zu diesem Directory-Eintrag an. Mit dem Action-Code NP werden alle im Verzeichnis enthaltenen Sicherungsversionen zu der Datei angezeigt. Für weitere Erläuterungen der Action-Codes wird auf das Kapitel 6 "Action-Codes" verwiesen.

Ausgewählte Datenobjekte: Jobvariablen

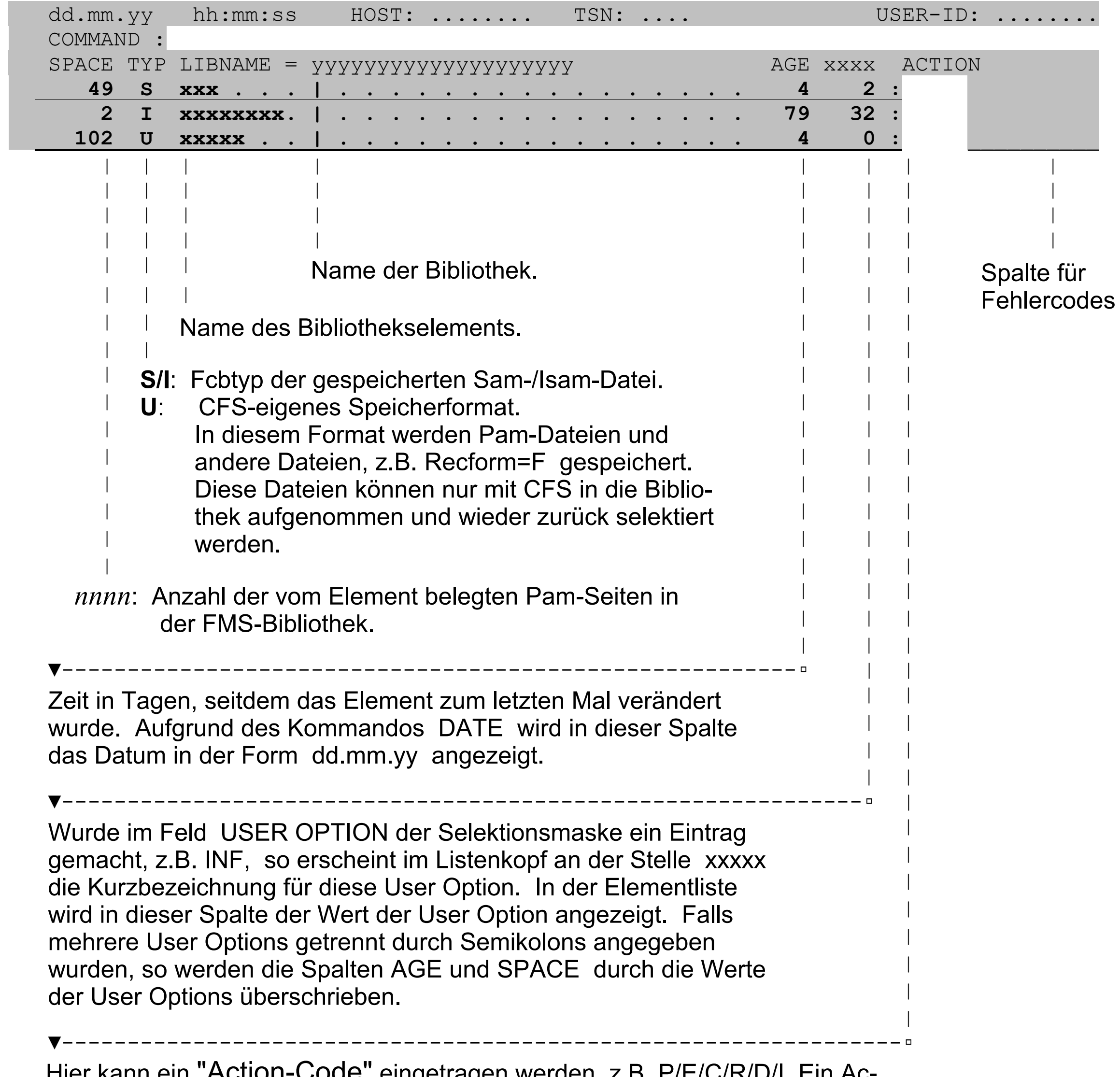


Ausgewählte Objekte: Elemente einer PLAM-Bibliothek



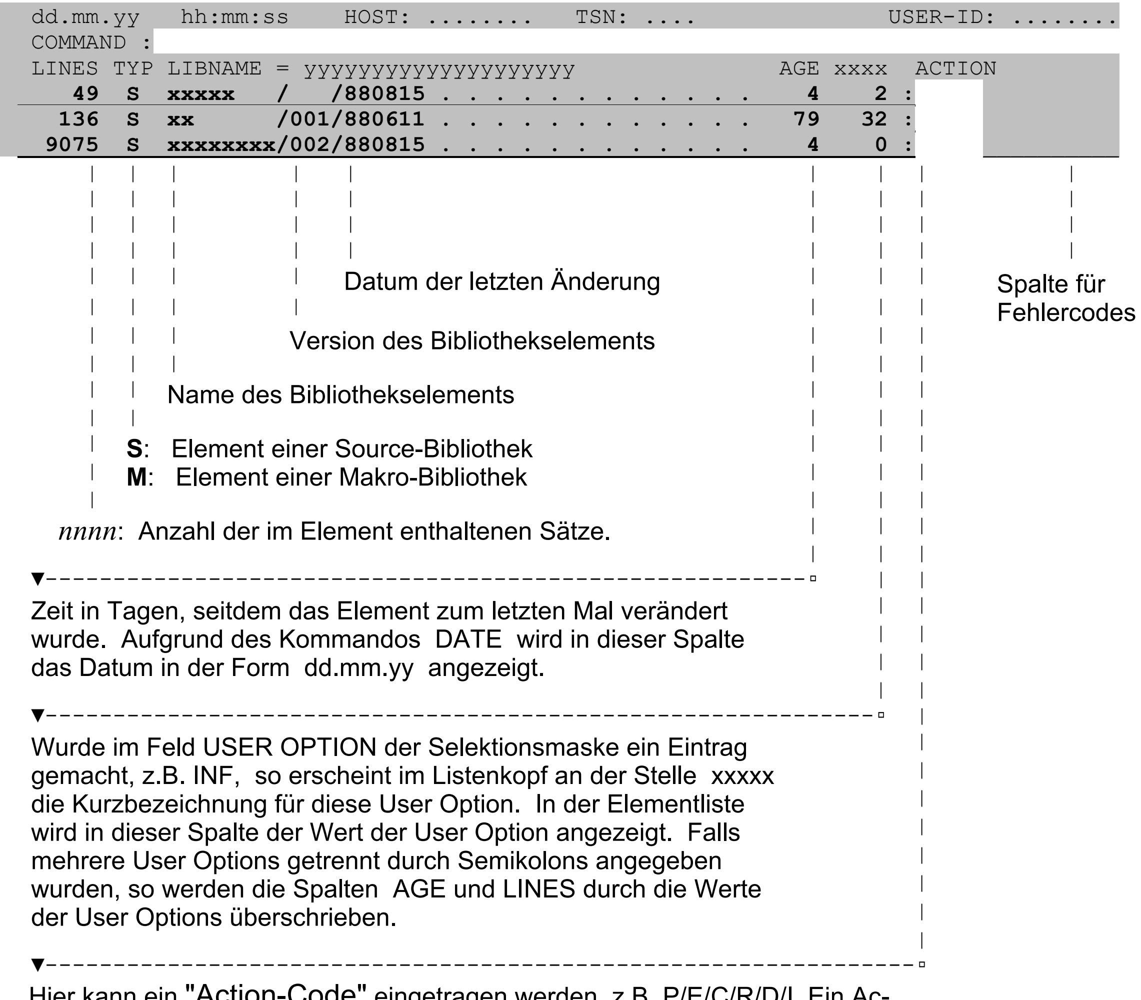
Hier kann ein "Action-Code" eingetragen werden, z.B. E/C/R/D/F/I. Ein Action-Code bewirkt eine bestimmte Aktion für das Element. Z.B. Erase/Copy/Rename/Display. Bestimmte Action-Codes werden sofort nach dem Absenden der Maske ausgeführt, z.B. Fstat/Copy/Rename. Andere Action-Codes werden gesammelt und am Ende der Verarbeitung gesammelt ausgeführt, z.B. Erase/Variable Actions. Für weitere Erläuterungen siehe Kapitel 6 "Action-Codes".

Ausgewählte Datenobjekte: Elemente einer FMS-Bibliothek



Hier kann ein "Action-Code" eingetragen werden, z.B. P/E/C/R/D/I. Ein Action-Code bewirkt eine Aktion für das Element, z.B. Print/Erase/Copy/ Rename/Display/Inform. Bestimmte Action-Codes werden sofort nach Absenden der Maske ausgeführt, z.B. Fstat/Inform/Copy/Rename. Andere Action-Codes werden gesammelt und am Ende der Verarbeitung ausgeführt, wie z.B. Print/Erase/Variable Actions. Für weitere Erläuterungen der Action-Codes wird auf das Kapitel 6 "Action-Codes" verwiesen.

Ausgewählte Datenobjekte: Elemente einer LMS-Bibliothek (OSM, ISAM)

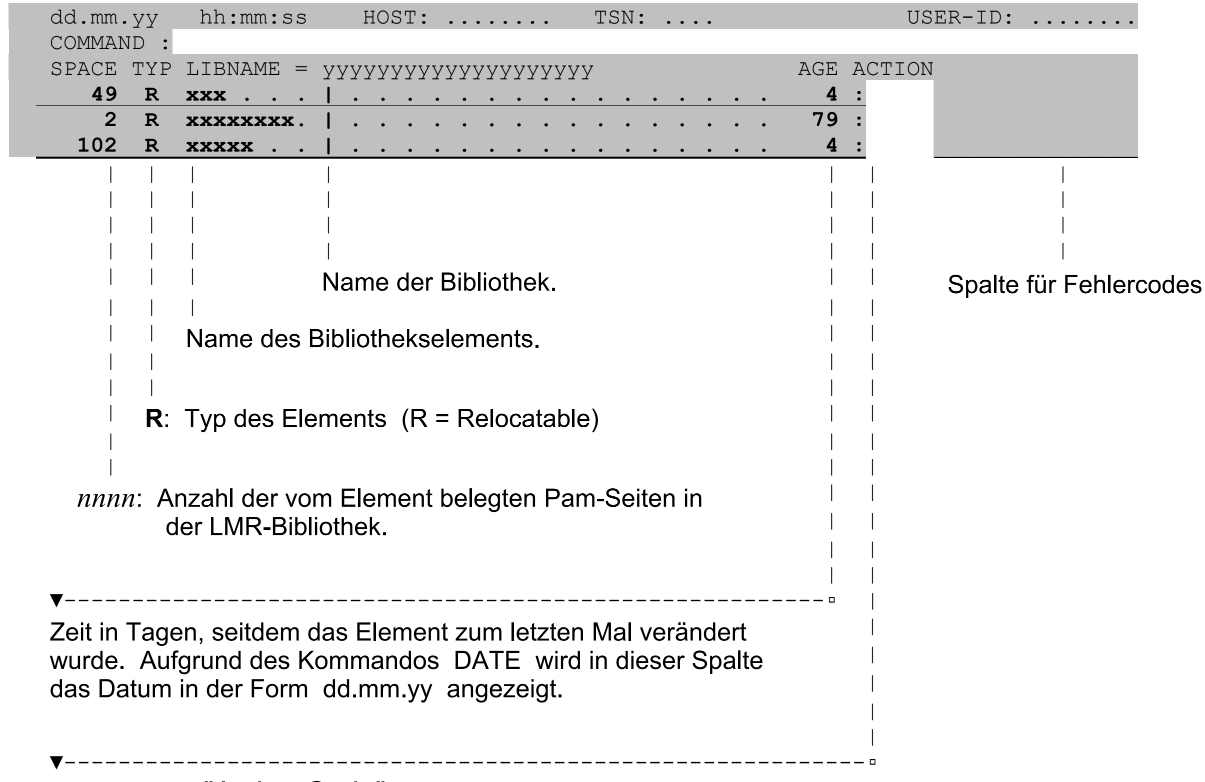

```
dd.mm.yy   hh:mm:ss    HOST: ........   TSN: ....            USER-ID: ........
COMMAND :
LINES TYP LIBNAME = yyyyyyyyyyyyyyyyyyyy                  AGE xxxx  ACTION
   49  S  xxxxx   /   /880815 . . . . . . . . . . .    4    2 :
  136  S  xx       /001/880611 . . . . . . . . . . .   79   32 :
 9075  S  xxxxxxxx/002/880815 . . . . . . . . . . .    4    0 :
```


Zeit in Tagen, seitdem das Element zum letzten Mal verändert wurde. Aufgrund des Kommandos DATE wird in dieser Spalte das Datum in der Form dd.mm.yy angezeigt.

Wurde im Feld USER OPTION der Selektionsmaske ein Eintrag gemacht, z.B. INF, so erscheint im Listenkopf an der Stelle xxxxx die Kurzbezeichnung für diese User Option. In der Elementliste wird in dieser Spalte der Wert der User Option angezeigt. Falls mehrere User Options getrennt durch Semikolons angegeben wurden, so werden die Spalten AGE und LINES durch die Werte der User Options überschrieben.

Hier kann ein "Action-Code" eingetragen werden, z.B. P/E/C/R/D/I. Ein Action-Code bewirkt eine bestimmte Aktion für das Element, z.B. Print/ Erase/ Copy/ Rename/ Display/ Inform. Bestimmte Action-Codes werden sofort nach Absenden der Maske ausgeführt, z.B. Copy/ Rename/ Inform. Andere Action-Codes werden gesammelt und am Ende der Verarbeitung ausgeführt, z.B. Print/ Erase/Variable Actions. Für weitere Erläuterungen der Action-Codes wird auf das Kapitel 6 "Action-Codes" verwiesen.

Ausgewählte Datenobjekte: Elemente einer LMR-Bibliothek



Hier kann ein "Action-Code" eingetragen werden. Ein Action-Code bewirkt eine bestimmte Aktion für das Element, z.B. Print/Display/Inform. Bei Elementen einer LMR-Bibliothek ist die Menge der zulässigen Action-Codes stark eingeschränkt. Es können nur die folgenden Action-Codes angegeben werden: D/I/P/S/U/X/*/+/-/-P.

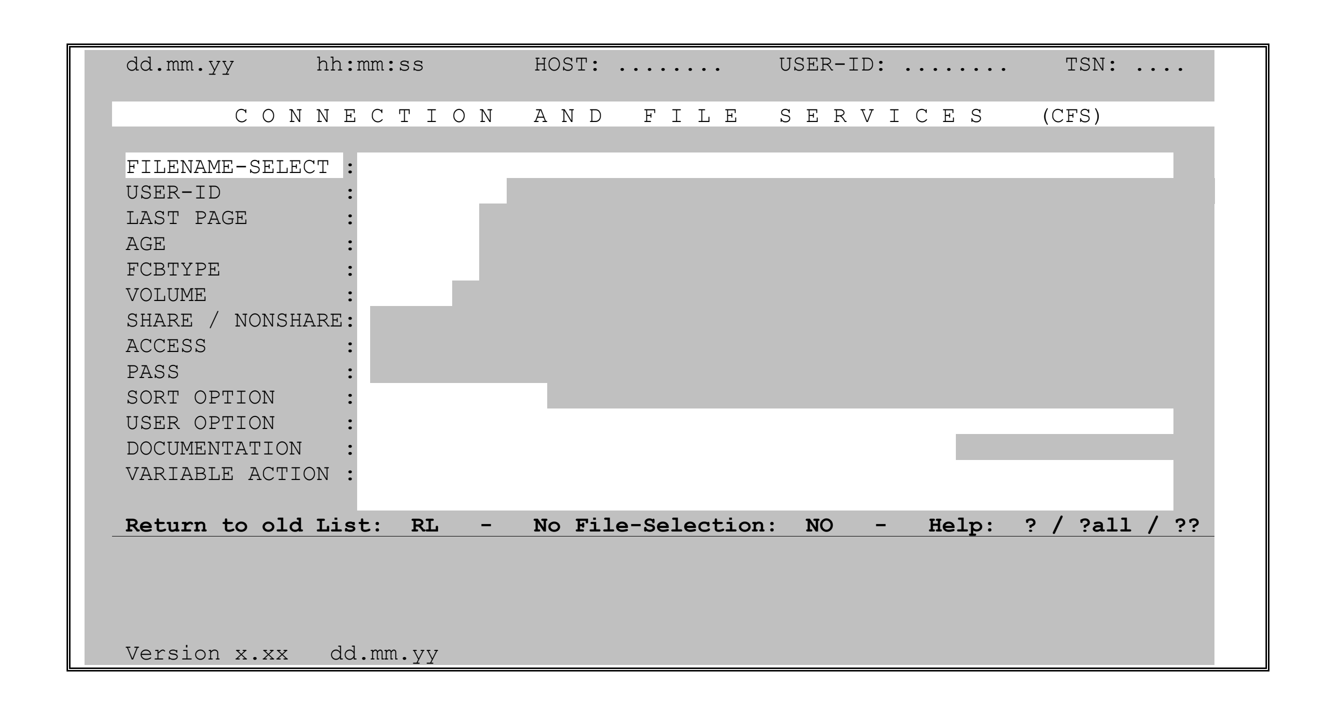
Display-Maske (CFS-Editor)

```
dd.mm.yy   hh:mm:ss    HOST: ........    TSN: ....             USER-ID:  ........
COMMAND :
LASTP TYP FILENAME = $TSOS                                  AGE PASS LACC  ACTION
   30 *I  xxxxxxxxxxxx  . . . . . . . . . . . . . . .  143 *       0 :
    1:zzzzzzzzzzzzzzzzzzz........................................................
    2:zzzzzzzzzzzzzzzzzzzz.......................................................
    3:zzzzzzzzzzzzzzzzzzzz.......................................................
    4:zzzzzzzzzzzzzzzzzzzzzzzzzzz................................................
    5:zzzzzzzzzzzzzzzzzzzz.......................................................
    6:zzzzzzzzzzzzzzzzzzzzz......................................................
    7:zzzzzzzzzzzzzzzzzzzzzz.....................................................
    8:zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz............................
    9:zzzzzzzzzzzzzzzzzzzzzzzzz..................................................
   10:zzzzzzzzzzzzzzzzzz.........................................................
   11:zzzzzzzzzzzzzzzzzzzzz......................................................
   12:zzzzzzzzzzzzzzzzzzzzzzzzzzz................................................
   13:zzzzzzzzzzzzzzzzzzzzzzzzzzzzz..............................................
   14:zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz......................................
   15:zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz.................................
   16:zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz......................................
   17:zzzzzzzz...................................................................
   18:zzzzzzzzzzzzzz.............................................................
   19:zzzzzzzzzzzzzzzzzzzzzzzzzz.................................................
Help:  ? / ?cmd/ ?all / ??
```

Die "Display-Maske" ist dem CFS-Editor zugeordnet. In dieser Maske wird der Inhalt der Datei/ der Jobvariablen/des Bibliothekselements angezeigt. Der Benutzer kann das Sichtfenster, wie vom EDT gewohnt, nach oben/unten/rechts/links verschieben. Er kann sich den Inhalt des Datenobjekts in verschiedenen Darstellungsweisen anzeigen lassen (z.B. Character- und Hexadezimalformat) und er kann den angezeigten Inhalt auch modifizieren.

Der CFS-Editor kennt keine Einschränkungen bezüglich des Dateiformats, der Satzlänge, sowie der Dateigröße.

Selektionsmaske : FILENAME-SELECT



```
dd.mm.yy      hh:mm:ss        HOST: ........    USER-ID: ........    TSN: ....

        C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -   No File-Selection:  NO   -   Help:  ? / ?all / ??



Version x.xx   dd.mm.yy
```

Auswahl von Dateien nach Merkmal im Namen

Einfache Auswahlbedingung

[*col*] [-] [*p*] '*string*'

Es werden alle Datenobjekte selektiert, deren Name einen bestimmten Suchbegriff enthält.

col :col1-col2: | :col: | >:col: | <:col:
Bei der Angabe eines Bereichs (:col: bzw. :col1-col2:) muß das angegebene Suchmuster '*string*' in diesem Spaltenbereich beginnen. Siehe hierzu auch Hinweis auf der nächsten Seite.

\- negative Auswahl: Es werden alle Datenobjekte selektiert, deren Name die Zeichenfolge '*string*' nicht enthält.

p > | <
Es werden alle Datenobjekte selektiert, deren Name eine Zeichenfolge größer/ kleiner '*string*' enthält.

Standard: = '*string*'

'*string*' Suchmuster.
Die Hochkommas zur Begrenzung des Suchmusters können im allgemeinen weggelassen werden. Selektionsmuster, die mit einem CFS-Schlüsselwort wie z.B. FMS/ LMS/ LMR MAC/ NO/ RL/ FSTAT/ STAJV/ RDTFT/ OLDLIST beginnen, müssen in Hochkommas eingeschlossen werden.

Hinweise:

Wird im Feld FILENAME-SELECT kein Suchbegriff angegeben, so werden alle Dateien /Jobvariablen ausgewählt, deren Eigenschaften den übrigen in der Selektionsmaske eingetragenen Suchkriterien entsprechen.

Durch Absenden der leeren Selektionsmaske werden alle Dateien der eigenen Benutzerkennung ausgewählt.

Bei der Angabe des Spaltenbereichs *col* in der Suchanweisung sind einige Unterschiede zur Spaltenbereichsangabe im EDT zu beachten:

1) :col1-col2: besagt, daß der in Hochkommas eingeschlossene Suchbegriff in dem angegebenen Spaltenbereich beginnen muß.
Im EDT muß der Suchbegriff vollständig im angegebenen Spaltenbereich enthalten sein.

2) :col: besagt, daß der in Hochkommas eingeschlossene Suchbegriff genau in der angegebenen Spalte beginnen muß.
Im EDT bewirkt die Angabe einer einzigen Spalte :col:, daß der Suchbegriff von Spalte :col: bis Satzende gesucht wird. In CFS wird dies durch die Angabe >:col2: erreicht, wobei col2=col-1 ist. Durch <:col: kann in CFS eine Zeichenfolge im Bereich vom Anfang des Namens bis zur Spalte col gesucht werden.

Beispiele:

`LST`
Dateiname muß die Zeichenfolge 'LST' enthalten.

`#`
Es werden **alle temporären Dateien** des aktuellen Prozesses ausgewählt.

`'#'`
Es werden alle regulären Dateien (keine Tempfiles), die das Zeichen '#' irgendwo im Namen enthalten ausgewählt.

`-'.'`
Name darf keinen Punkt '.' enthalten.

`:1:<'D'`
Name muß mit einem der Buchstaben A, B oder C (Zeichen <'D') beginnen.

`:1-3:-'.'`
Name darf nicht mit einer 1- bis 3-stelligen Teilqualifizierung beginnen.

`:9:' '`
Name muß in Spalte 9 ein Leerzeichen enthalten, d.h. der Name darf nur 1 bis 8 Stellen lang sein.

`->'Z'`
Name darf keine Ziffer (kein Zeichen > 'Z') enthalten.

Mehrfachauswahl - Format 1: *param* [*vk param*] [*vk param*] .....

param einfaches Suchargument gemäß der oben beschriebenen Syntax.

vk Verknüpfung mit dem vorausgegangenen einfachen Suchargument.

, Oder-Verknüpfung.

+ Und-Verknüpfung.

***** [*n*] Wildcard-Verknüpfung: Und-Verknüpfung, jedoch muß das zweite Suchitem im Dateinamen nach dem ersten Suchitem vorkommen. Falls der Wert *n* angegeben wurde, müssen zwischen dem ersten und dem zweiten Suchargument genau *n* beliebige Zeichen stehen.

Es können beliebig viele Suchargumente durch Oder-/Und-/Wildcard-Bedingungen verknüpft werden.

Beispiele:

```
CFS*SRC
```
Alle Dateien, deren Name die Zeichenfolge 'CFS' und irgendwo danach 'SRC' enthält.

```
:1:>'B'+:1:<'H',:1:'X',:1:'Y',:1:'Z'
```
Alle Dateien, die mit den Buchstaben C, D, E, F, G, X, Y oder Z beginnen.

Mehrfachauswahl - Format 2: ( ) | (?)

Durch () wird eine spezielle Such-Maske angefordert. Dort stehen dem Benutzer 18 Bildschirmzeilen zur Aufnahme von umfangreichen Suchargumenten zur Verfügung. Für eine Beschreibung der Such-Maske siehe Seite **315**.
Durch (?) wird die mit den zuletzt verwendeten Suchbegriffen ausgefüllte Such-Maske angezeigt. (?) ist die analoge Funktion zu S? im Kommandofeld von CFS.

Mehrfachauswahl - Format 3: %*name* | %?

%*name* ist die Bezeichnung eines Parameter-Sets, unter dem zu einem früheren Zeitpunkt ein komplexes Suchargument gespeichert wurde (SP %*name*).
%?: Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden. Näheres hierzu siehe Seite **315**.

Mehrfachauswahl - Format 4: (*s-dat*)

s-dat ist der Name einer Datei (SAM), in der die Suchargumente festgehalten sind. Näheres hierzu siehe Seite **316**.

Beispiel:

```
LMS CFS.TESTPLAM,(CFS.SEL.1)
```

In der SAM-Datei CFS.SEL.1 seien folgende Datensätze gespeichert:

```
'OUT'+-'LOGOUT'+'A1'
'CFS'*'SRC'
:1:'S/'
```

Die angegebenen Selektionssätze werden von CFS zu folgender Suchanweisung zusammengesetzt: 'OUT'+-'LOGOUT'+'A1','CFS'*'SRC',:1:'S.'

Aufgrund dieser Anweisung werden alle Elemente in der Bibliothek CFS.TESTPLAM selektiert, deren Name eine der nachstehenden Bedingungen erfüllt:

- Elementname enthält die Zeichenfolgen 'OUT' und 'A1', jedoch nicht 'LOGOUT'
- Elementname enthält die Zeichenfolge 'CFS' und irgendwo dahinter die Zeichenfolge 'SRC'.
- Elementname beginnt mit der Zeichenfolge 'S/' (Elementtyp S).

Auswahl von Jobvariablen

Die Selektion von Jobvariablen kann durch Eingabe von J im Feld FCBTYPE der Selektionsmaske gesteuert werden. Im Feld FILENAME-SELECT können nach der oben dargestellten Syntax beliebige Auswahlbedingungen bezüglich des Namens eingegeben werden.

STAJV | SHJVA Alle Jobvariablen einer Benutzerkennung oder alle durch eine Teilqualifizierung eingeschränkten Jobvariablen können auch über das Feld FILENAME-SELECT selektiert werden durch die Eingabe von STAJV [*teilqual.*] bzw. SHJVA [*teilqual.*].

Auswahl von Bibliothekselementen

FMS *bibl* | **LMS** *bibl* | **LMR** *bibl* | **MAC** *bibl*

Die "Dateienliste" besteht aus allen Elementen der angegebenen Bibliothek. Keine Auswahl nach einem Suchmuster im Namen der Elemente.

bibl Name der Bibliothek, deren Elemente aufgelistet werden sollen.

FMS *bibl , param* | **LMS** *bibl , param* | **LMR** *bibl , param* | **MAC** *bibl , param*

Die Dateienliste besteht nur aus denjenigen Elementen der angegebenen Bibliothek, deren Namen das in *param* spezifizierte Selektionskriterium erfüllen.

param einfache oder zusammengesetzte Auswahlbedingung.
Syntax siehe Seite **49** (Auswahl von Dateien nach Merkmal im Namen).

Hinweise:

Mit dem Schlüsselbegriff 'LMS' können sowohl Bibliotheken des alten LMS-Formats (LMS-ISAM, 8-stelliger Elementname), als auch Bibliotheken des neuen LMS-Formats (PLAM, 44-stelliger Elementname) ausgewählt werden. Von LMR erzeugte OMLs werden durch das Schlüsselwort LMR ausgewählt.

Anstelle der Schlüsselbegriffe FMS/LMS/MAC können auch die Langformen FMSLIB/ LMSLIB/ LMRLIB/ MACLIB angegeben werden.

PLAM: CFS-intern werden die Namen der in PLAM-Bibliotheken gespeicherten Elemente in der Form *t/element/vers* geführt. Bei Elementen mit der Standard-Versionsbezeichnung (ß, X'FF') wird diese am Bildschirm nicht angezeigt. In LMS wird die Standard-Versionsbezeichnung durch das Zeichen @ dargestellt.

LMR: LMR-Bibliotheken können von CFS nur im Input-Modus eröffnet werden. Hieraus ergeben sich Einschränkungen bezüglich der Menge der zur Verfügung stehenden Variablen Actions und Action-Codes. Für eine LMR-Bibliothek stehen z.B. die Action-Codes M (Modify), E (Erase), C (Copy) und R (Rename) nicht zur Verfügung.

Mit der Variablen Action ONXADD *bibl* können Module einer LMR-Bibliothek in eine PLAM-Bibliothek übertragen werden.

Beispiele:

`FMS EST.FMSLIB`
Es wird das gesamte Inhaltsverzeichnis der FMS-Bibliothek EST.FMSLIB angezeigt.

`LMS MACROLIB,>:10:'001'`
Selektion aller Elemente der LMS-Bibliothek MACROLIB mit einer Versionsnummer > 001. LMS-ISAM: Versionsnummer beginnt fest in Spalte 10.

`LMS BIBL.PROG,X/`
Alle Elemente der PLAM-Bibliothek BIBL.PROG mit dem Elementtyp X.
Elemente mit einem bestimmten Elementtyp können auch durch Eintragen des gewünschten Elementtyps im Feld FCBTYPE selektiert werden.

`LMS BIBL.PROG,:3:'ABC'`
Selektion aller Elemente der PLAM-Bibliothek BIBL.PROG, deren Name mit der Zeichenfolge ABC beginnt. Die Spaltenangabe :3: ist notwendig, da in CFS ein Name in der Form S/ABC.TEST/002 dargestellt wird.

`LMS BIBL.PROG,/ß`
Selektion aller Elemente in der PLAM-Bibliothek BIBL.PROG mit der Standardversion X'FF'.

`LMS BIBL.PROG,/TEST/`
alle Elemente der PLAM-Bibliothek BIBL.PROG mit dem vollqualifizierten Namen TEST. Bei dieser Variante der Namensselektion spielt der jeweilige Elementtyp und die Elementversion keine Rolle.

Auswahl von Archive-Directory Einträgen

DIR *directory* [, *param*]

Die Dateienliste zeigt alle Einträge für die eigene Kennung aus dem angegebenen Archive-Directory bzw. nur diejenigen Einträge, deren Namen die in *param* angegebene Suchbedingung erfüllen.

directory Name des Archive-Directories, dessen Einträge aufgelistet werden sollen.

param einfache oder zusammengesetzte Auswahlbedingung bezüglich der Namen der im Archive-Directory enthaltenen Einträge. Syntax siehe Seite **49** (Auswahl von Dateien nach Merkmal im Namen).

Hinweise:

Standardmäßig werden die Einträge für alle CAT-IDs der eigenen Kennung angezeigt. Im Feld USER-ID der Selektionsmaske kann die Auswahl auf Einträge eines bestimmten PVS eingeschränkt werden (Eingabe :*pvs*:).

Standardmäßig wird für eine Datei/Jobvariable nur der letzte (neueste) Sicherungseintrag angezeigt. Durch zusätzliche Angabe der USER OPTION: SAVE bzw. SAVE CNS werden alle Sicherungseinträge (FULL/PART) bzw. zusätzlich auch die CNS-Einträge (Catalogued but not Saved) in der Liste angezeigt.
Mit dem Action-Code NP, eingegeben bei einem Eintrag in der Standard Directory-Liste, kann jederzeit ein Verzeichnis aller vorhandenen Sicherungseinträge angefordert werden.

Durch Eingabe von J im Feld FCBTYPE der Selektionsmaske werden alle Jobvariablen im Archive-Directory angezeigt.

Für Archive-Directory Einträge stehen auch die User Options RETPD und SAVE zur Verfügung. Für nähere Informationen siehe Seite **99**/**100**.
Zur Anzeige von Volume-Informationen aus dem Archive-Directory steht das Kommando POOL zur Verfügung. Näheres siehe Seite **259**.

Beispiele:

`DIR $TSOS.ARCHIVE.SICHERUNG`
Es werden die letzten Sicherungsstände aller Dateien der eigenen Benutzerkennung aus dem Archive-Directory $TSOS.ARCHIVE.SICHERUNG angezeigt.

`DIR $TSOS.ARCHIVE.SICHERUNG,'LIB'`
Es werden die letzten Sicherungsstände derjenigen Dateien der eigenen Benutzerkennung angezeigt, deren Namen den String 'LIB' enthält.

Auswahl von Elementen eines ZIP-Archivs

ZIP *archiv* [, *param*]

Die Dateienliste zeigt alle Elemente des angegebenen ZIP-Archivs bzw. nur diejenigen Elemente, deren Namen die in *param* angegebene Suchbedingung erfüllen.

archiv Name des ZIP-Archivs, dessen Einträge aufgelistet werden sollen.

param einfache oder zusammengesetzte Auswahlbedingung bezüglich der Namen der im Archive-Directory enthaltenen Einträge. Syntax siehe Seite **49** (Auswahl von Dateien nach Merkmal im Namen).

Hinweis:

Als weitere einschränkende Selektionsmöglichkeiten für ZIP-Archive sind nur Einträge in den Feldern AGE und LASTP der Selektionsmaske möglich. Es existieren keine User Options für ZIP-Archive.

Beispiele:

`ZIP TMP.ZIPFILE`
Es werden alle in dem ZIP-Archiv TMP.ZIPFILE gespeicherten Elemente angezeigt.

`ZIP TMP.ZIPFILE,SAVE`
Es werden alle in dem ZIP-Archiv TMP.ZIPFILE gespeicherten Elemente angezeigt, die in ihrem Namen den String 'SAVE' enthalten.

Sonderoptionen im Feld FILENAME-SELECT

Dateiselektion über teil-/vollqualifizierten Namen

string , Q

Qualified Selection. Es wird genau die durch *string* vollqualifizierte Datei ausgewählt. Falls *string* mit einem Punkt endet, werden alle Dateien ausgewählt, deren Name mit der Teilqualifizierung *string.* beginnt.

Hinweise:

Die Q-Option ist ein einfaches Mittel zur teilqualifizierten und schnellen Selektion von Dateien/Jobvariablen.

Ohne Q-Option kann eine teilqualifizierte Dateiselektion nur über :1:'*string.*' bzw. FSTAT *string.* erreicht werden.

Dateiauswahl über FSTAT/SHFA-Kommando

FSTAT *xxx* [, *param*]

xxx Datei-Selektionsmuster gemäß FSTAT-Syntax (auch Wildcards möglich).

par zusätzliche FSTAT-Auswahlparameter (z.B. SIZE=30:300).
Der Operand LIST= des FSTAT-Kommandos darf nicht angegeben werden, da dieser intern von CFS verwendet wird.

SHFA *xxx* [, *param*] | **SHFA** ?

xxx Datei-Selektionsmuster gemäß SHFA-Syntax (auch Wildcards möglich).
? Es wird in den geführten SDF-Dialog des Kommandos SHFA (SHOW File Attributes) verzweigt.

par zusätzliche Auswahlparameter des Kommandos SHFA (z.B. SELECT=(SIZE=*INT(FROM=30,TO=300)).
Der Operand OUTPUT= des SHFA-Kommandos darf nicht angegeben werden, da dieser intern von CFS verwendet wird.

Beispiele:

```
FSTAT *CFS*DO*
SHFA *CFS*DO*
FSTAT VOL=PVT002,VTOC=YES
SHFA SELECT=(VOL=PVT002,FROM-CAT=*PRIV(PVT002,D3475))
```

Hinweise:

Im Normalfall werden die Cat-IDs und User-IDs der selektierten Dateien in der Dateienliste nicht ausgegeben. Das Kommando CFN (Complete Filenames), siehe Seite 470 bewirkt jedoch, daß die Namen der ausgewählten Dateien mit Cat-ID und User-ID angezeigt werden.

Bei der Selektion von Dateien über das FSTAT-bzw. SHFA-Kommando werden in der Dateienliste die Cat-ID's der ausgewählten Dateien aufgeführt. Außerdem können Dateien über alle Pubsets (PVS) selektiert werden: FSTAT/SHFA :*:.. Das gleiche Ergebnis kann auch durch Eingabe von :*: im Feld USER-ID erreicht werden.

Falls vor dem Laden von CFS mit RFASTART eine Verbindung zu einem entfernten Rechner hergestellt wurde, so kann durch FSTAT :*: bzw. FSTAT :x: auch eine Selektion von Dateien des fremden Rechners erfolgen. Die Selektion von Dateien eines RFA-Rechners über die FSTAT-Option ist schneller als die Selektion über das Feld USER-ID (Eingabe *:x:*). Die Selektion wird zusätzlich beschleunigt, falls im Feld USER OPTION : NO (Names Only) eingegeben wird.

Temporäre Dateien können durch Eingabe von # oder FSTAT # im Feld FILENAME-SELECT ausgewählt werden. Der Tempfile-Indikator '#' kann aufgrund optionaler BS2000-REPs auf das Alternativzeichen '@' gesetzt sein und ist dann als solcher anzugeben.

Temporäre Jobvariablen können durch Eingabe von # oder STAJV # im Feld FILE-NAME-SELECT ausgewählt werden.

Ist die Ausgabe der Cat-IDs in der Dateienliste nicht erwünscht (z.B. bei Sicherungen mit der Variablen Action ONXARCHIVE), kann dies durch Eintragen von NO (NoCat-IDs) im Feld USER OPTION erreicht werden.

Falls in einer Benutzerkennung sehr viele Dateien verwaltet werden, so kann bei der Selektion von Dateien dieser Kennung von CFS folgende Meldung ausgegeben werden: "FSTAT-Area too small. Some Files will be missing in the File-List". Diese Engpaß-Situation kann durch Auswahl der gewünschten Dateien über die FSTAT-Option vermieden werden.

Alle mit FT zu übertragenden Dateien selektieren (NSTAT)

NSTAT

Es werden alle Dateien/PLAM-Bibliothekselemente selektiert, die mit openFT zu oder vom eigenen Host übertragen werden und deren Übertragung noch nicht beendet ist.

Hinweise:

Für erweiterte Optionen unter TSOS (z.B. NSTAT *ALL), geben Sie ein: ?all und dann Menuepunkt "Funktionen für den Systemverwalter"

Der Aufbau der Dateienliste unterscheidet sich bei dieser Option vom Aufbau der Trefferliste für Dateien und ist auf Seite **41** beschrieben. Der Action-Code U bewirkt in der NSTAT-Liste, daß der aktuelle Status des File-Transfer Auftrags in der Liste aktualisiert wird. Bei einem laufenden File-Transfer Auftrag wird z.B. die Anzahl der momentan übertragenen Bytes in K (Kilobytes) angezeigt.

Bei Anwendung der Option NSTAT können keine weiteren Selektionsmerkmale in der Selektionsmaske eingetragen werden. Im Gegensatz zu FSTAT können auch keine Operanden des BS2000-Kommandos NSTAT angegeben werden.

Zum Thema File-Transfer mit openFT siehe auch Variable Action ONXFT Seite **143**, Action-Codes FT/FTS/FTC/FTCF Seite **194**, Kommando FT Seite **241** und Kommandos JRFT/NJRFT Seite **412**.

Auswahl aller Dateien mit LINK-Namen (RDTFT)

RDTFT

Es werden alle Dateien selektiert, die im aktuellen Prozeß mit einem Link-Namen verknüpft sind. Die Link-Namen werden in der Dateienliste mit den Dateinamen zusammen angezeigt.

Die Benutzerkennung der ausgewählten Dateien spielt bei dieser Selektionsform keine Rolle. Es werden auch Dateien aus anderen Benutzerkennungen angezeigt, falls diese mit dem aktuellen Prozeß über einen Link-Namen verknüpft sind. Falls für eine

Datei mehrere Link-Namen existieren, so wird sie mit den entsprechenden Link-Namen auch mehrfach in der Dateienliste aufgeführt.

Im Zusammenhang mit der RDTFT-Option wird auch auf den Action-Code RL (Release Link) hingewiesen. Mit diesem Action-Code kann ein in der Dateienliste aufgeführter Linkname freigegeben werden.

Keine Datenobjekte auswählen

NO [; *cmd*]
Es findet keine Selektion irgendwelcher Datenobjekte statt. In der als nächstes angezeigten Maske (Dateienliste) wird lediglich das Kommandofeld (COMMAND) von CFS ausgegeben.

cmd
Automatische Eingabe im Feld COMMAND der Dateienliste. Der angegebene String wird als Kommando interpretiert und sofort ausgeführt.

Hinweis:

Auch in den Zusatzkomponenten JES/JESG/TAS wird über die Eingabe NO;*cmd* im ersten Feld der Selektionsmaske eine leere Objektliste ausgewählt und im Feld COMMAND der String *cmd* zur Ausführung gebracht.

Beispiel:

`NO;OC1 S43,CFSSAP1,,SAP`
Es erfolgt keine Auswahl von Datenobjekten. Im Feld "COMMAND" der zweiten CFS-Maske wird das Kommando OC1 S43,CFSSAP1,,SAP ausgeführt und damit eine Connection zu der SAP-Anwendung S43 eröffnet.

`NO;TAS;`
Im Feld "COMMAND" der zweiten CFS-Maske wird das Kommando TAS; ausgeführt. Das Kommando TAS; ist gleichbedeutend mit dem leeren Absenden der Selektionsmaske von TAS und bewirkt die Auswahl aller zur Zeit aktiven Tasks in einer Liste.

`NO;TASNO;DSSM`
Abweichend von dem oben dargestellten Aufruf der CFS-Komponente TAS werden in diesem Beispiel keine Tasks ausgewählt (TASNO). Im Feld "COMMAND" der leeren Taskliste wird das Kommando DSSM ausgeführt. Damit werden alle im Subsystemkatalog enthaltenen Subsysteme und deren Status ausgegeben.

Auswahl über Parameterset

%*name* | %? [, *jrsave* | ?]

Die Selektion wird mit den in dem Parameterset gespeicherten Angaben durchgeführt.

%*name*
Name des Parametersets mit der Selektionsbedingung.

%?
Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

jrsave
Name der JRSAVE-Datei, in der der Parameterset gespeichert ist.
Standard: CFS.JRSAVE.*xxxx*. Näheres siehe Seite 565.

?
Die Selektionsmaske wird mit dem Inhalt des Parametersets gefüllt und dem Benutzer angezeigt. Standardmäßig wird die Selektion sofort ausgeführt.

Hinweis:

Das Abspeichern einer Selektionsbedingung erfolgt in der Dateienliste von CFS mit dem Kommando SP %*name*. Näheres hierzu auf Seite 283.

Frühere Dateienliste wieder herstellen

RL [*mn*] Restore List.

RL Es wird die zuletzt ausgewählte Dateienliste angezeigt. Diese Option kann nicht verwendet werden, falls der ADOC-Modus (Automatic Doc-File) ausgeschaltet ist (Standard: ADOC eingeschaltet).

RL *mn* Es wird die mit dem Kommando SL *mn* weggeschriebene Dateienliste angezeigt.

Der RL-Mechanismus ist eine verkürzte Form der OLDLIST-Option (siehe unten).

Frühere Dateienliste anzeigen

OLDLIST *datei* [, *param*] Die Dateienliste wird erzeugt aus den Namen, die enthalten sind in:

a) CFS-Dokumentationsdatei. Eine Dokumentationsdatei wird erzeugt durch Eintragen eines Dateinamens im Feld DOCUMENTATION in der Selektionsmaske. Eine Dokumentationsdatei kann auch mit den Kommandos DOC und SL (Save List) erzeugt werden.

b) Referenzdatei. Eine Referenzdatei ist eine beliebige SAM-Datei, bei der jeder Satz aus einem Dateinamen besteht. Eine Referenzdatei kann z.B. auf folgende Art und Weise erzeugt werden:

- Durch ein FSTAT-Kommando der Form:
  /FSTAT ....,LIST=*datei* bzw. /FSTAT ....,LIST=(*datei*,F)
  Die in der List-Datei enthaltenen Überschriften (FSTAT ...,LIST=*datei*) werden von CFS überlesen.

- Durch die Kommandofolge:
  /SYSFILE SYSLST=*datei*
  /IMPORT VOLUME=*volume*,DEVICE=*device*,LIST=(ONLY,SYSLST)
  /SYSFILE=()
  Die in der List-Datei enthaltenen Überschriften werden von CFS überlesen.

- Im EDT mit @FS 1 und @W'*datei*

- Mit der Variablen Action DPF von CFS: ONXDPF *datei*

param Durch Angabe eines zusätzlichen Suchmusters *param* bzw. durch Eintragen zusätzlicher Auswahlbedingungen in den anderen Feldern der Selektionsmaske ist es möglich, eine zwei-, bzw. mehrstufige Selektion durchzuführen.

Beispiel:

```
FILENAME-SELECT : OLDLIST datei,:1:'ABC'
AGE             : >60
```

selektiert alle diejenigen Dateien aus der angegebenen Dokumentationsdatei, deren Name mit der Zeichenfolge 'ABC' beginnt und die seit mindestens 60 Tagen nicht mehr verändert wurden.

Hinweise:

Durch Eintragen der User Option: NO (Names Only) zusammen mit der OLDLIST-Option wird beim Aufbau der Dateienliste nicht geprüft, ob die in der Referenzdatei aufgeführten Datenobjekte tatsächlich existieren. Damit können in der Dateienliste Namen von Objekten ausgegeben werden, die von CFS nicht unterstützt werden. Beispiel: Liste der Namen aller NK-ISAM Pools. Eine CFS-Prozedur hierzu finden Sie unter dem Namen NKISAM-POOLS in der Bibliothek CFS.USERLIB. BEISPIEL.

Zum Thema "frühere Dateienlisten wieder herstellen" siehe auch Kommandos SL/RL (Save List/Restore List) Seite **275**/**281**, ADOC (Automatic Documentation) Seite **468** und DOC Seite **473**, sowie Feld DOCUMENTATION in der Selektionsmaske Seite **74**.

Anwendungsbeispiel für die OLDLIST-Option

Abholen von Dateien von einem fremden Rechner mit openFT.

Mit FSTAT ...,LIST=*datei* wird auf dem Remote-Rechner eine Namensliste von Dateien erzeugt (Referenzdatei). Diese Datei wird anschließend auf den eigenen Rechner übertragen (NCOPY-Kommando). Durch

```
FILENAME-SELECT : OLDLIST datei
USER OPTION     : NO
```

wird auf dem eigenen Host die auf dem Remote-System erstellte Namensliste als Dateienliste ausgewählt. Mit der Variablen Action ONXFT *rem-host*,*user-id*,*acct*, *passw*,,,,FROM können nun die gewünschten Dateien durch Ankreuzen mit dem Action-Code X in das eigene System übertragen werden.

Gedächtnis der Eingaben in der Selektionsmaske

CFS führt eine interne Tabelle, in der alle in der Selektionsmaske getätigten Eingaben aufgezeichnet werden. Auf dieses "Gedächtnis" kann der Benutzer auf zwei verschiedene Arten zugreifen:

a) sequentiell:

Durch Betätigen der Tasten Shift ENTER (DUE2) bei leerem Feld FILENAME-SELECT wird die letzte, vorletzte, vorvorletzte usw. Eingabe angezeigt.

b) assoziativ:

string Shift ENTER. Es wird die letzte Eingabe in der Selektionsmaske angezeigt, die im Feld FILENAME-SELECT mit dem angegebenen Suchmuster beginnt. Durch weiteres Betätigen der Tasten Shift ENTER (DUE2) wird die vorletzte Eingabe angezeigt usw.

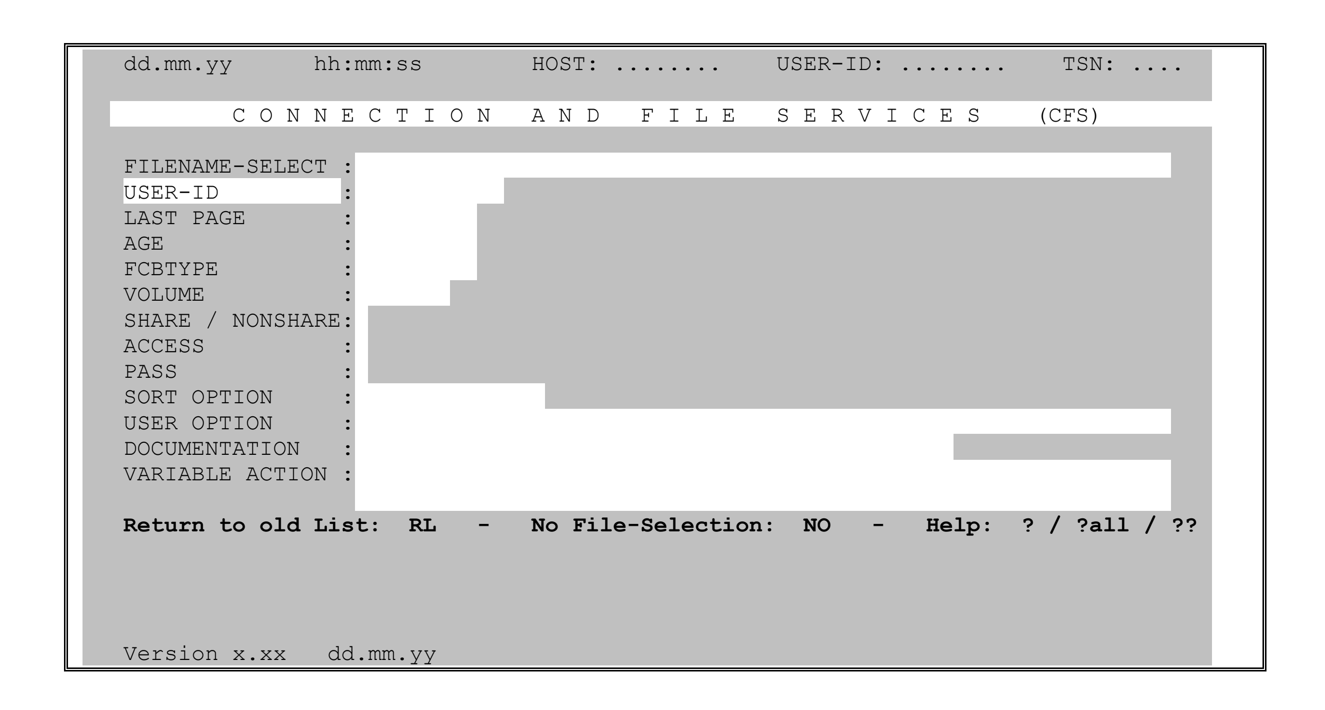
**string* Shift ENTER. Es wird die letzte Eingabe in der Selektionsmaske angezeigt, die an irgendeiner Stelle das Suchmuster '*string*' enthält.

Beispiele:

`FILENAME-SELECT : LMS` Shift ENTER (DUE2)
zeigt die letzte Selektionseingabe, die mit LMS beginnt.
z.B. LMS CFS.PLAMLIB,-R/.

`FILENAME-SELECT : *PLAM` Shift ENTER (DUE2)
zeigt die letzte Selektionseingabe, die an irgendeiner Stelle die Zeichenfolge 'PLAM' enthält. z.B. LMS CFS.PLAMLIB.

Selektionsmaske : USER-ID



```
dd.mm.yy      hh:mm:ss        HOST: ........     USER-ID: ........    TSN: ....

        C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -   No File-Selection:  NO   -   Help:  ? / ?all / ??



Version x.xx   dd.mm.yy
```

Benutzerkennung / PVS für Dateiauswahl

:*x*: | :*[*]: | *user-id* | :*x*: *user-id* | :*[*]: *user-id*

:*x*: — Selektion von Dateien bzw. Archive-Directory Einträgen im PVS (Public Volume Set) *x* der eigenen Benutzerkennung.

:*: — Selektion von Dateien in allen lokal verfügbaren Pubsets (PVS) der eigenen Benutzerkennung.

:**: — Selektion von Dateien in allen lokal und remote zugreifbaren Pubsets (PVS) der eigenen Benutzerkennung.

user-id — Selektion von Dateien im Default-Pubset der angegebenen Benutzerkennung. Die Benutzerkennung ist ohne $ und nachgestellten Punkt anzugeben. Falls die Benutzerkennung dennoch mit einem führenden '$' eingegeben wurde, so wird das erste $-Zeichen der User-ID ignoriert.

:*x*: *user-id* — Selektion von Dateien im Pubset *x* der angegebenen Benutzerkennung.

:*: *user-id* — Selektion von Dateien in allen lokal verfügbaren Pubsets der angegebenen Benutzerkennung.

:**: *user-id* — Selektion von Dateien in allen lokal und remote zugreifbaren Pubsets der angegebenen Benutzerkennung.

Bei leer gelassenem Feld User-Id werden Dateien im Default-Pubset der eigenen Benutzerkennung selektiert.

Hinweise:

Der Benutzer kann eine User-Id als Standard vordefinieren. Ausführliche Informationen hierzu siehe Kommando UID, Seite 484.

Bis BS2000 Version 9.5 gilt folgendes: Unter einer Benutzerkennung ungleich TSOS werden nur die nicht shareable Dateien aus einer fremden Kennung angezeigt.
Ab BS2000 Version 10.0 sind die erweiterten Zugriffsschutzeinrichtungen BACL (Basic Access Control List) und ACL (Access Control List) implementiert. Dies hat zur Folge, daß auch nicht shareable Dateien aus fremden Kennungen angezeigt werden können, sofern die BACL-Schutzeinrichtungen für die Dateien aktiviert sind und einen Zugriff (Lesend/Schreibend/Ausführend) erlauben. Umgekehrt können Dateien aus fremden Kennungen die shareable sind, nicht angezeigt werden, falls die BACL-Attribute einen Zugriff für die aktuelle Kennung nicht zulassen.

Beispiele:

`:B:USER2`
Benutzerkennung $USER2 im Katalog :B:

`:B:`
Public-Volume-Set :B: unter der eigenen Benutzerkennung.

`$$USER3`
Die Benutzerkennung beginnt selbst mit einem $-Zeichen: $USER3.
Das erste $-Zeichen in der Benutzerkennung wird von CFS ignoriert.

Dateien / Jobvariablen über mehrere Kennungen auswählen

() Die USER-IDs werden aus der Datei CFS.USERID.COOWNER unter der eigenen Kennung, bzw. aus einer mit /FILE ...,LINK=USERID zugewiesenen Datei gelesen. Wie der Name sagt, sind in der Datei alle User-IDs aufgeführt, für die die eigene Kennung die Coownerschaft besitzt. Die eigene Kennung sollte als erste enthalten sein.

(*datei*) Die USER-IDs werden aus der angegebenen Datei gelesen. Die Datei muß die Eigenschaften Recform=V und Fcbtype=SAM besitzen.

Beispiele:

Die Datei CFS.USER enthält 4 Datensätze mit folgendem Inhalt:

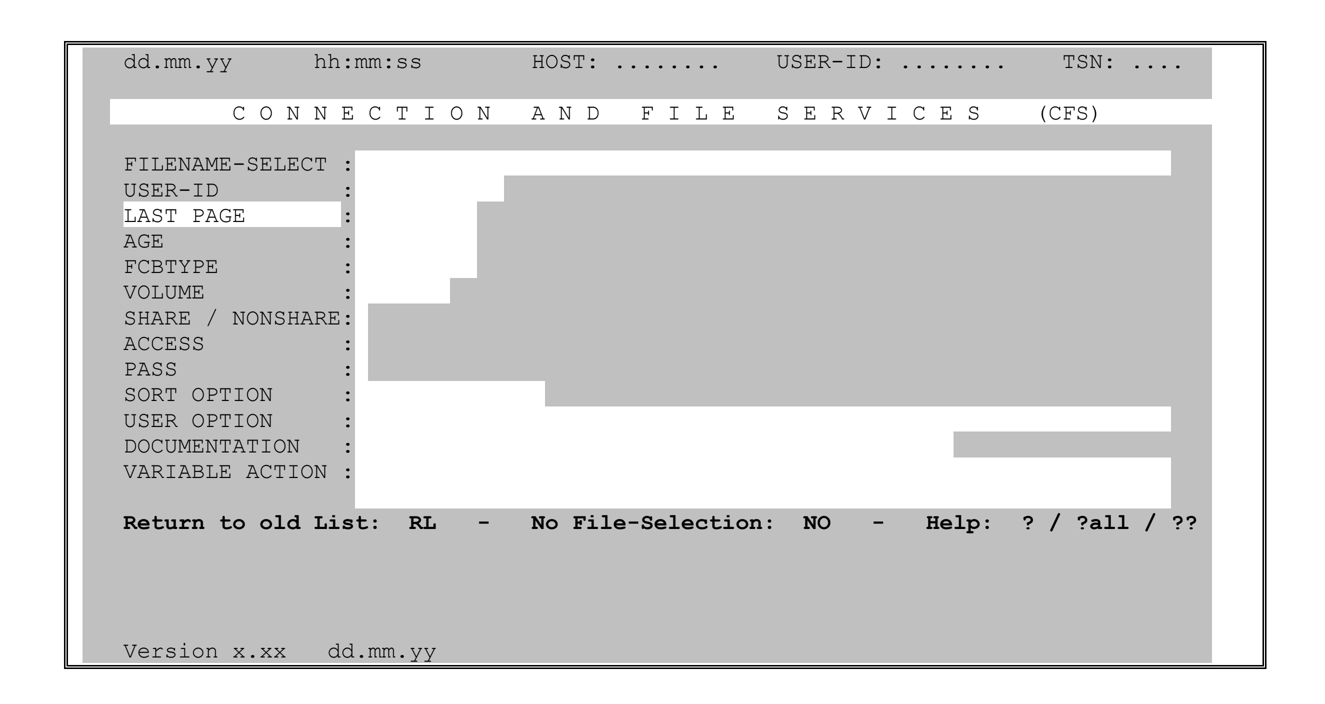
```
AA10
CFS
TEST4711
TSOS
```

`NPTMP;(CFS.USER)`
Es werden alle Dateien ausgewählt, die in ihrem Namen die Zeichenfolge 'TMP' enthalten und die unter den Default Cat-IDs der Benutzerkennungen $AA10., $CFS., $TEST4711. oder $TSOS. katalogisiert sind.

`NPTMP;:*:(CFS.USER)`
gleiche Selektion wie oben, aber es werden jetzt alle Cat-IDs der der jeweiligen Benutzerkennungen berücksichtigt.

(SYSDTA) Die gewünschten USER-IDs werden von SYSDTA angefordert/eingelesen.
Es erscheint am Bildschirm ein Stern '*' als Eingabeaufforderung. Nach der letzten User-ID ist als Endekennzeichen 'EOF' anzugeben oder die K1-Taste zu betätigen.

Selektionsmaske : LAST PAGE


```
dd.mm.yy      hh:mm:ss        HOST: ........    USER-ID: ........    TSN: ....

        C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -   No File-Selection:  NO   -   Help:  ? / ?all / ??



Version x.xx   dd.mm.yy
```

Auswahl nach Dateigröße

>*x* | **=***x* | **<***x*

Es werden Dateien/Bibliothekselemente mit einer Größe von mehr als (>), weniger als (<), oder genau (=) *x* beschriebenen PAM-Seiten ausgewählt.

Hinweise:

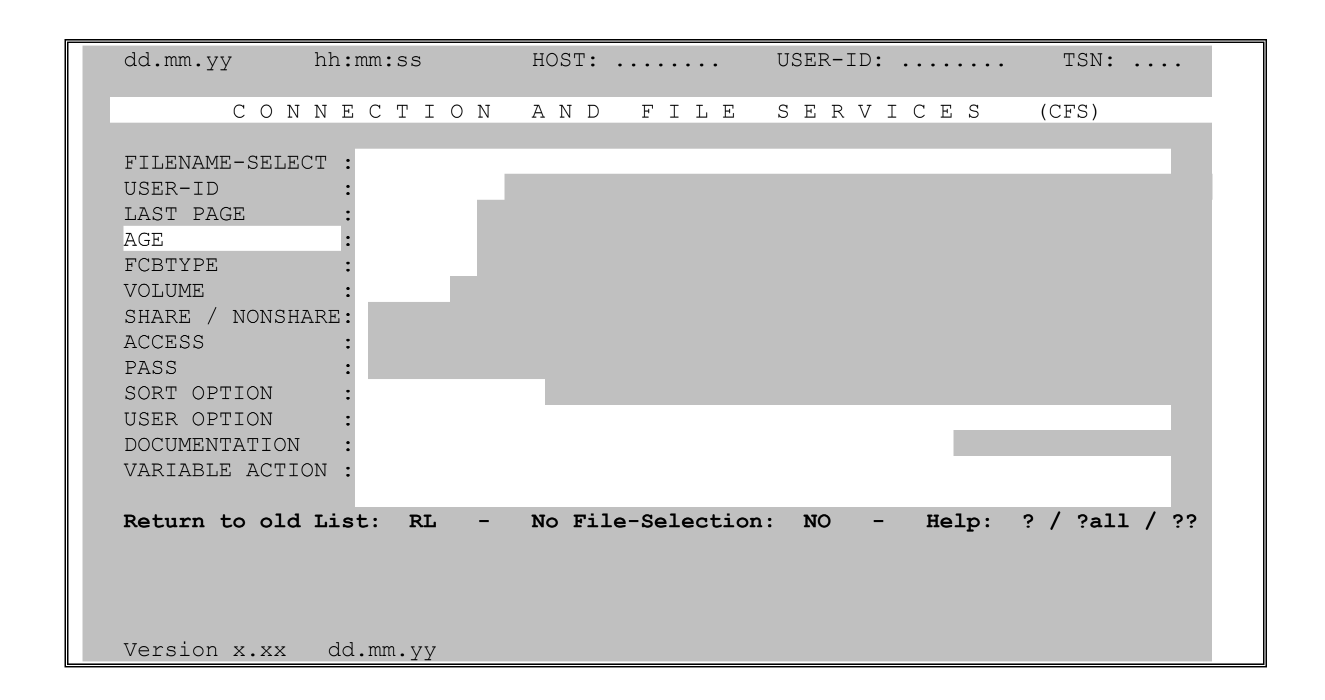
Das Zeichen '=' kann bei der Selektionsangabe auch weggelassen werden. Die Angabe einer bloßen Zahl *x* wird in diesem Fall als =*x* interpretiert.

Bei Selektion von Jobvariablen (FCBTYPE : J) kann im Feld Last Page bezüglich der Anzahl der beschriebenen Datenbytes der Jobvariablen selektiert werden.

Beispiele:

<4	Alle Dateien mit weniger als 4 beschriebenen Seiten.
0	Alle leeren Dateien (0 beschriebene PAM-Seiten).

Selektionsmaske : AGE



```
dd.mm.yy      hh:mm:ss        HOST: ........    USER-ID: ........     TSN: ....

        C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -   No File-Selection:  NO   -   Help:  ? / ?all / ??



Version x.xx   dd.mm.yy
```

Auswahl nach Alter / Datum

Es werden Datenobjekte (Dateien/Bibliothekselemente/Jobvariablen) ausgewählt, die vor mehr als (>), weniger als (<), oder genau (=) *x* Tagen bzw. vor/nach dem angegebenen Datum erstellt/zum letzten Mal verändert wurden.

Auswahl nach Alter:	>*x* | =*x* | <*x* (0 - 15000)
Auswahl nach Datum:	>*dd.* [*mm.* [*yy*]] | <*dd.* [*mm.* [*yy*]] | =*dd.* [*mm.* [*yy*]]

Hinweise:

Bei Dateien und Jobvariablen erfolgt die Selektion aufgrund des BS2000-Katalogmerkmals "Change Date".

Bei PLAM-Bibliothekselementen wird standardmäßig das im Inhaltsverzeichnis als User-Date gespeicherte Datum zur Age-Berechnung verwendet. Soll das System-Datum als AGE-Kriterium verwendet werden, so ist vor der Selektion das Kommando SDAT (siehe Seite 481) anzugeben. Das System-Datum entspricht dem Change Date bei Dateien.

Anstelle der Selektionsangabe =*x* kann auch der einfache Wert *x* angegeben werden. Das Zeichen '=' wird in diesem Fall von CFS ergänzt.

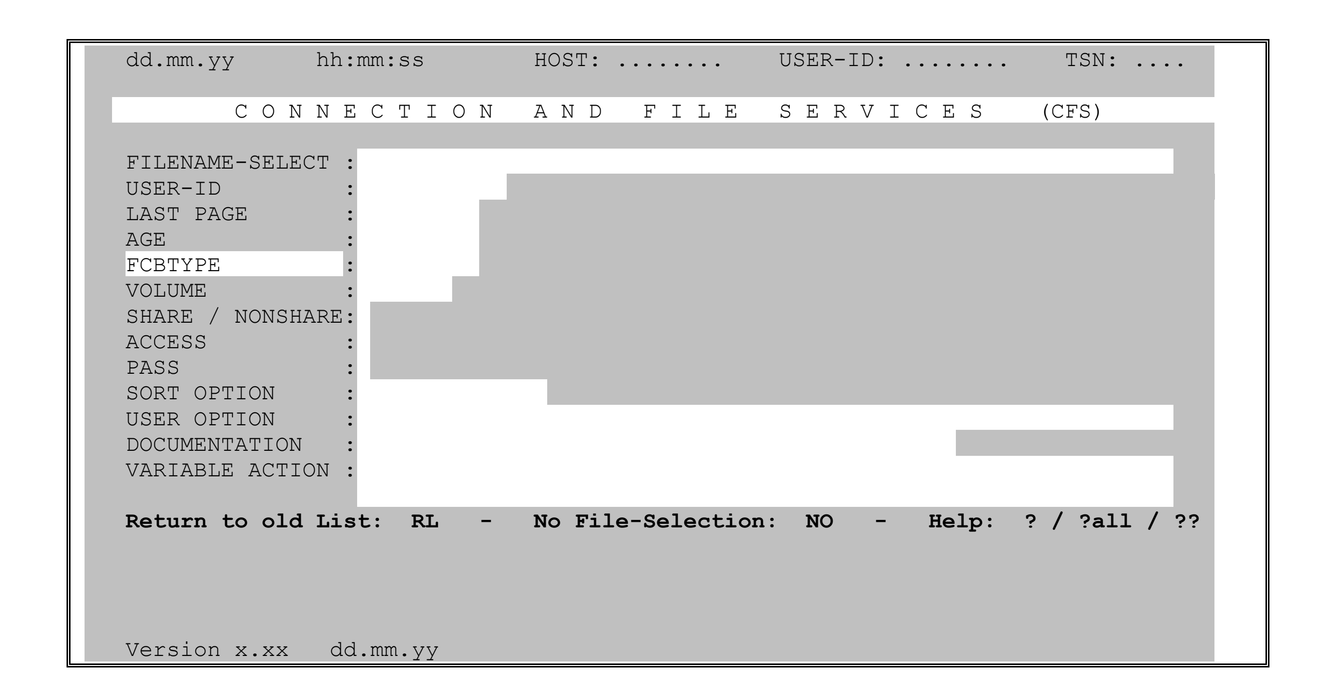
Falls im Datum *dd.mm.yy* die Monats-/ Jahresangaben (*mm.yy* / *yy*) weggelassen wurden, so werden intern die aktuellen Monats- und Jahresangaben ergänzt. Es ist jedoch zu beachten, daß das letzte Zeichen ein Punkt sein muß.

Mit dem Kommando DATE (Seite 473) wird veranlaßt, daß das Alter der Datenobjekte nicht als Zeitangabe in Tagen, sondern in Form eines Datums angezeigt wird.

Beispiele:

`0`	Alle heute erstellten bzw. veränderten Dateien/Bibliothekselemente/Jobvariablen.
`>7`	Es werden Datenobjekte ausgewählt, die vor mehr als 7 Tagen verändert bzw. erstellt wurden.
`<15.05.`	Es werden Datenobjekte ausgewählt, die nach dem 15.05. des aktuellen Jahres das letzte Mal verändert wurden.

Selektionsmaske : FCBTYPE


```
dd.mm.yy       hh:mm:ss        HOST: ........    USER-ID: ........    TSN: ....

        C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -   No File-Selection:  NO   -   Help:  ? / ?all / ??



Version x.xx   dd.mm.yy
```

Auswahl nach FCBTYP

Alle Dateien mit einem bestimmten FCBTYP/PLAM-Bibliotheken/Jobvariablen/alle Elemente einer PLAM-Bibliothek mit einem bestimmten Element-Typ.

Dateien:

[-] S	[nicht] SAM
[-] I	[nicht] ISAM
[-] P	[nicht] PAM
[-] B	[nicht] BTAM
[-] N	[nicht] NONE (NONE = mit File- bzw. CAT-Kommando erzeugte und noch nicht eröffnete Dateien)
D	SAM oder ISAM
[-] PL	[nicht] Programmbibliotheken (PLAM)

Jobvariablen:

J	Selektion von Jobvariablen bzw. Jobvariablen Einträge aus einem Archive-Directory.
JM	Selektion von Monitor-Jobvariablen

Selektionsmaske : FCBTYPE

Elemente einer PLAM-Bibliothek:

[-] C	[nicht] Coreimage (gebundene Programme)
[-] D	[nicht] Daten
[-] H	[nicht] von H-Assembler erzeugte Daten
[-] J	[nicht] Job-Control (Enter-Dateien/DO-Prozeduren)
[-] L	[nicht] von Binder des BS2000 erzeugte Daten
[-] M	[nicht] Makro
[-] P	[nicht] Print (druckaufbereitete Listen)
[-] R	[nicht] Relocatable (Objektmodule)
[-] S	[nicht] Source (Quellprogramme)
[-] X	[nicht] Extra (Elementtyp zur freien Verfügung des Benutzers)

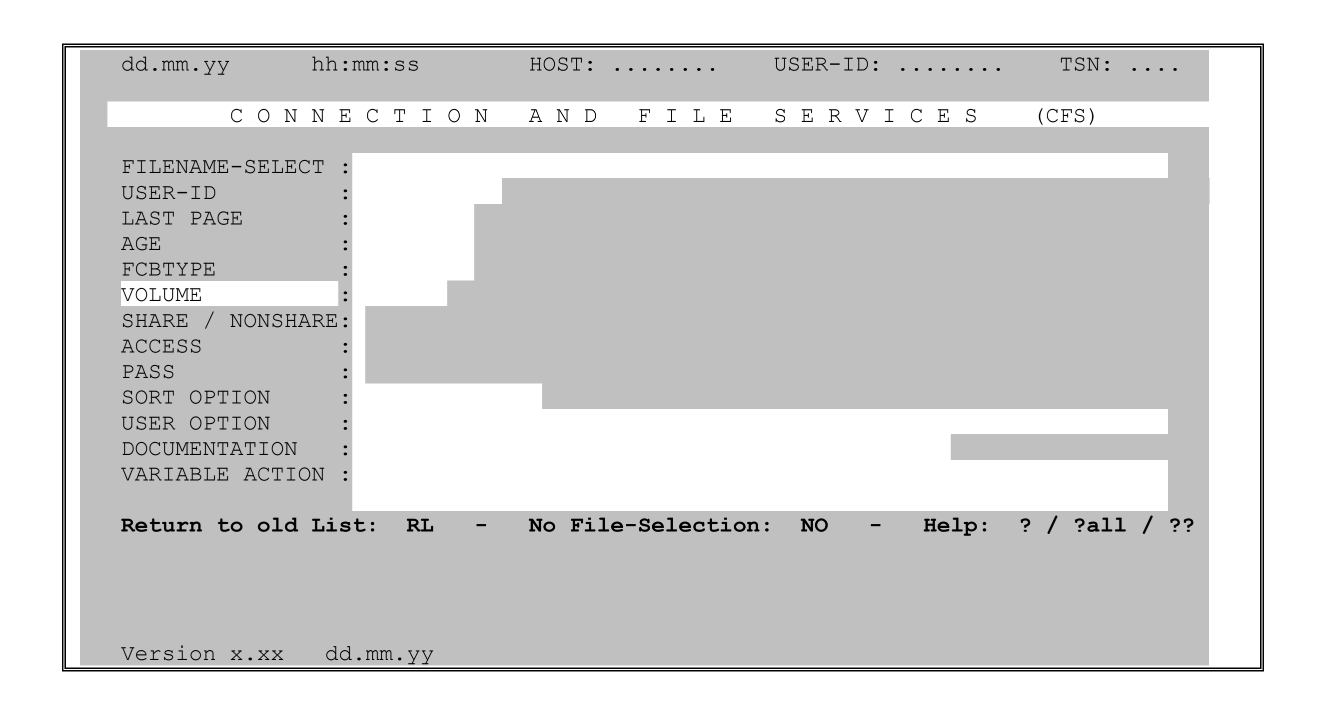
Hinweis:

Bei großen PLAM-Bibliotheken mit Elementen verschiedener Typen kann der Selektionsvorgang durch Auswahl eines bestimmten Elementtyps im Feld FCBTYPE beschleunigt werden.
Zum gleichen Thema siehe auch User Option NO (Names Only) Seite 92.

Elemente einer FMS-Bibliothek:

[-] S	[nicht] SAM
[-] I	[nicht] ISAM
[-] U	[nicht] Undefined (Freies Satzformat)

Selektionsmaske : VOLUME


dd.mm.yy hh:mm:ss HOST: .... USER-ID: .... TSN: ....

C O N N E C T I O N A N D F I L E S E R V I C E S (CFS)

FILENAME-SELECT :
USER-ID :
LAST PAGE :
AGE :
FCBTYPE :
VOLUME :
SHARE / NONSHARE:
ACCESS :
PASS :
SORT OPTION :
USER OPTION :
DOCUMENTATION :
VARIABLE ACTION :

Return to old List: RL - No File-Selection: NO - Help: ? / ?all / ??

Version x.xx dd.mm.yy

Auswahl von Dateien auf einem Datenträger/Public-Volume-Set

PUBLIC Dateien auf gemeinschaftlichen Datenträgern.

PRDISK Dateien auf Private-Volumes (Privatplatten).

NET Dateien, die in einem NET-STORAGE Dateisystem gespeichert sind.

TAPE Katalogeinträge für Banddateien.
-TAPE Alle Dateien außer Banddateien.

FGG Nur Dateigenerationsgruppen (Indexeinträge + Generationen).
-FGG Keine Dateigenerationen und dazugehörige Indexeinträge.

:x: Alle Dateien auf dem angegebenen Public-Volume Set.

vsn Alle Dateien, die auf der Platte mit der angegebenen VSN (Public/Private) Speicherplatz belegt haben bzw. die einen Eintrag für diese vsn im Katalog haben.

-*vsn* Dateien, die auf der Platte mit der angegebenen VSN keinen Speicherplatz belegt haben.

*vsn*T Alle Dateien, die auf einem Magnetband mit der angegebenen Volume Serial Nummer (VSN) gespeichert sind.
Bei der Option *vsn*T muß die Volume-Serial-Number 6-stellig angegeben werden (evtl. mit führenden Nullen).

VSN mit Wildcards

/ steht für ein beliebiges Zeichen an der entsprechenden Stelle der vsn (analog FSTAT-Kommando).

Beispiele:

`//A///` Alle Dateien, die auf Magnetplatten Speicherplatz belegt haben, deren VSN an der dritten Stelle das Zeichen 'A' enthält.

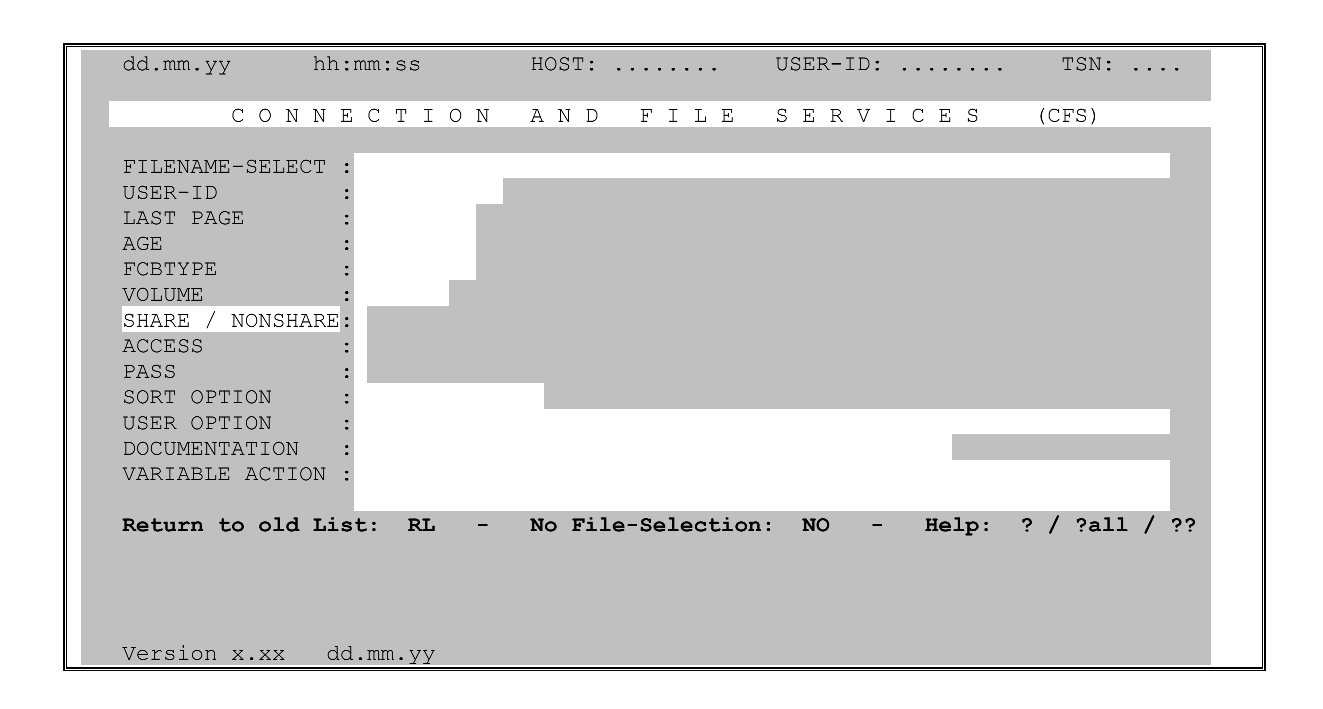
`-PVT///` Alle Dateien, die nicht auf einem Datenträger gespeichert sind, dessen VSN mit der Zeichenfolge 'PVT' beginnt.

`1/////T` Alle Dateien auf Magnetbändern, deren Volume Serial Nummern an der ersten Stelle eine 1 enthalten.

Hinweis:

Falls sich eine Datei auf einem privaten Datenträger befindet (Privatplatte/ Magnetband), so wird dies in der Dateienliste durch das Zeichen ***** in der ersten Stelle der Spalte LASTP angezeigt.

Selektionsmaske : SHARE / NONSHARE



```
dd.mm.yy      hh:mm:ss        HOST: ........    USER-ID: ........    TSN: ....

        C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -   No File-Selection:  NO   -   Help:  ? / ?all / ??




Version x.xx   dd.mm.yy
```

Auswahl bezüglich SHARE-Attribut

S Dateien / Jobvariablen mit der Eigenschaft SHARE=YES
Share=Yes Dateien können auch durch die Angabe Y ausgewählt werden.

N Dateien / Jobvariablen mit der Eigenschaft SHARE=NO

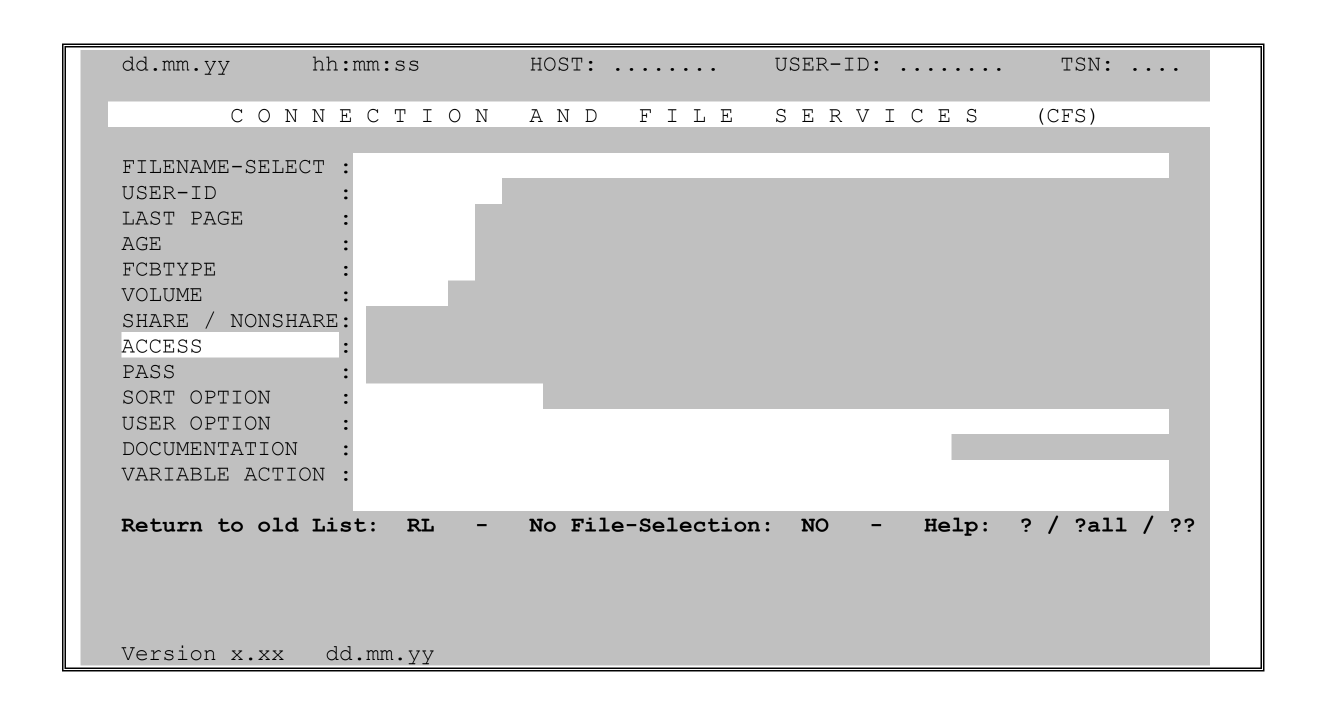
X Dateien mit der Eigenschaft SHARE=SPECIAL
SHARE=SPECIAL schließt SHARE=YES ein, und bewirkt, daß die Dateien auch von der Benutzerkennung $SERVICE aus zugreifbar sind.

Hinweise:

Die Eigenschaft SHARE=YES wird in der Dateienliste durch das Zeichen ***** vor dem Fcbtyp dargestellt.

Die Eigenschaft SHARE=SPECIAL wird in der Dateienliste durch das Zeichen **#** vor dem Fcbtyp dargestellt.

Selektionsmaske : ACCESS



```
dd.mm.yy       hh:mm:ss        HOST: ........    USER-ID: ........    TSN: ....

        C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -   No File-Selection:  NO   -   Help:  ? / ?all / ??



Version x.xx   dd.mm.yy
```

Auswahl bezüglich ACCESS-Merkmal

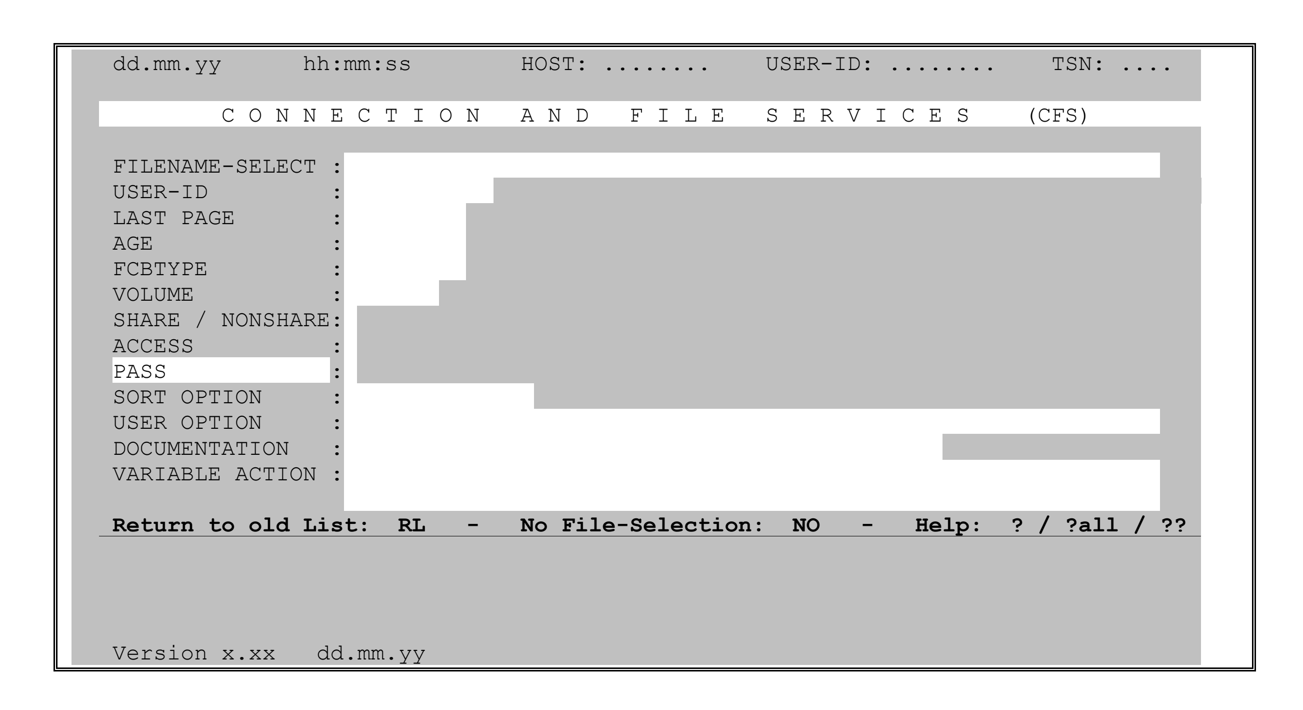
R Dateien / Jobvariablen mit der Eigenschaft ACCESS=READ

W Dateien / Jobvariablen mit der Eigenschaft ACCESS=WRITE

Hinweis:

Die Eigenschaft ACCESS=READ wird in der Dateienliste durch das Zeichen * in der Spalte PASS dargestellt.

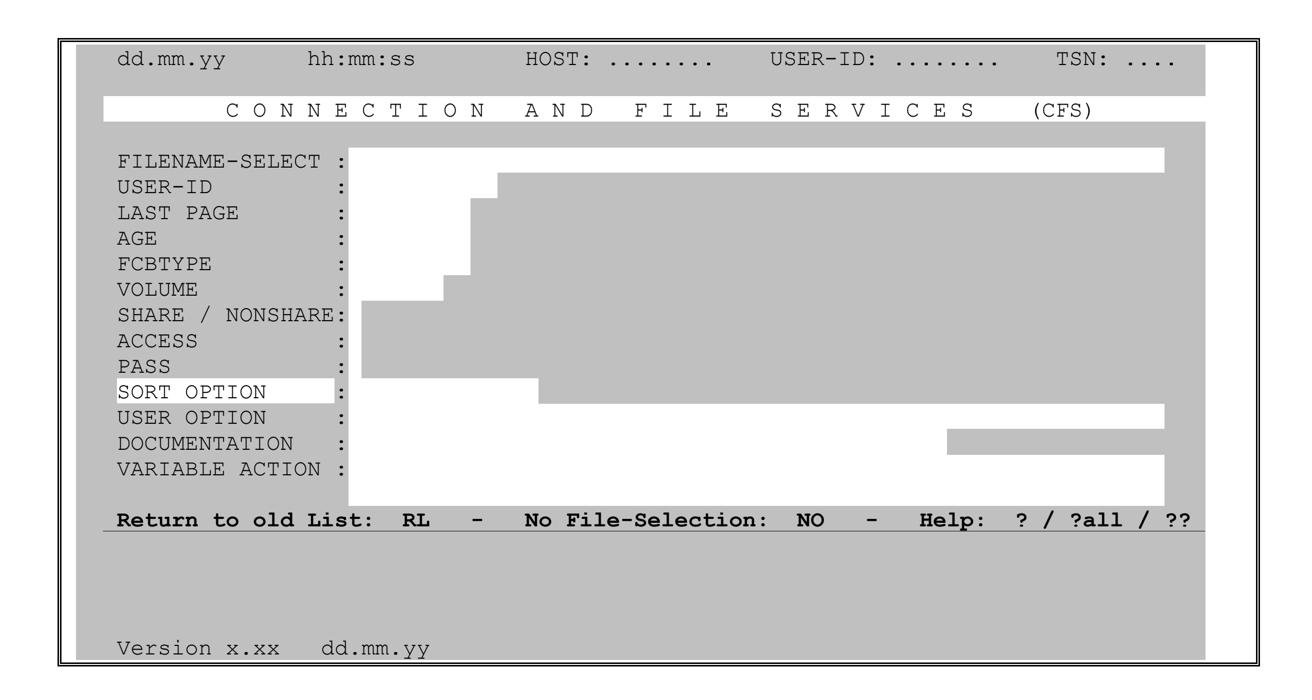
Selektionsmaske : PASS



Auswahl von Dateien / Jobvariablen / PLAM-Bibliothekselementen mit einem bestimmten Typ von Passworten.

W	Dateien / Jobvariablen / PLAM-Elemente mit einem WRITE-Passwort.
R	Dateien / Jobvariablen / PLAM-Elemente mit einem READ-Passwort.
E	Dateien / PLAM-Elemente mit einem EXEC-Passwort.
X | Y	Dateien / Jobvariablen / PLAM-Elemente mit irgendeinem Passwort.
N	Dateien / Jobvariablen / PLAM-Elemente ohne Passwort.

Selektionsmaske : SORT OPTION



Sortierkriterium für Dateienliste bestimmen

Die Dateienliste wird standardmäßig nach Namen aufsteigend sortiert. Durch die SORT OPTION kann ein anderes Feld als Sortierkriterium angegeben werden. Daneben kann auch die Reihenfolge der Sortierung (auf-/absteigend) bestimmt werden. Felder mit dem gleichen Sortierkriterium werden in zweiter Instanz nach Namen aufsteigend sortiert.

(*c*,*l*) | AGE | NAME [(*c*,*l*)] | SIZE | USER | NONE [, A | D]

(*c*,*l*) Sortierung nach einer absoluten Spaltenangabe (*c* = Column, *l* = Länge).
Die Spaltenangabe bezieht sich auf die Spalten eines in der Dateienliste eingeblendeten Zeilenlineals (Kommando SC).
Durch die Angabe (8,1) kann z.B. nach der Typ-Angabe sortiert werden.

AGE Die Dateienliste wird gemäß dem Inhalt in der AGE-, bzw. DATE-Spalte sortiert. Bei PLAM-Bibliotheken wird die Uhrzeit der letzen Änderung (TIME) im Rahmen der Sortierung mit berücksichtigt.

NAME [(*c*,*l*)] Die Dateienliste wird bezüglich der Namen der Datenobjekte sortiert (Standard). Über den Zusatz (*c*,*l*) kann ein Teil des Namens als Sortierfeld festgelegt werden. *c* gibt dabei die Spalte innerhalb des Namens und *l* die Länge des Namensteils an.
Standard: NAME(1,44).

Hinweis:

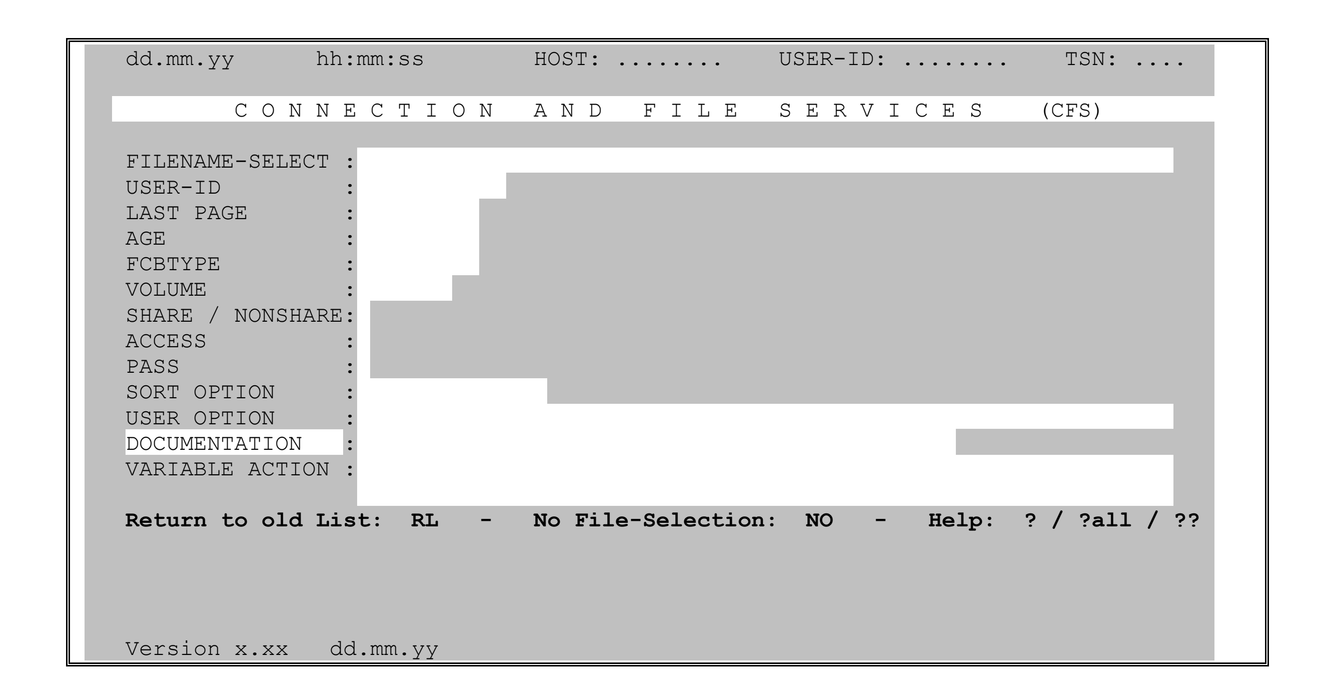
Bei PLAM-Bibliotheken kann durch die Sort Option NAME(1,44) eine Sortierung erreicht werden, bei der Elemente mit gleichen Namen und verschiedenen Typ-Bezeichnungen aufeinander folgen.
Durch die Sort Option NAME ohne Zusatz wird bei PLAM-Elementen der Elementtyp bei der Sortierung der Elementnamen mit berücksichtigt. Es ergibt sich in diesem Fall ein anderes Sortierergebnis als bei der Angabe NAME(1,44).

SIZE	Die Dateienliste wird gemäß dem Inhalt der ersten Spalte (LASTP, bzw. ALLOC bei Dateien und SPACE bei Bibliotheken) sortiert.
USER	Falls bei der Selektion der Dateienliste eine USER OPTION angegeben wurde, so wird durch die Sortieroption USER eine Sortierung gemäß dem in der User Option-Spalte enthaltenen Wert vorgenommen. Falls keine USER OPTION angegeben wurde, so bleibt diese Sortieroption ohne Wirkung.
NONE	Die Dateien-/Elementliste wird unsortiert ausgegeben.
A	Aufsteigende Sortierreihenfolge (Standard).
D	Absteigende Sortierreihenfolge.

Hinweis:

Das Feld SORT OPTION wird durch die Standard Sortierreihenfolge vorbelegt, falls diese von NAME,A abweicht. Die Standard Sortierreihenfolge kann mit dem Kommando SO (siehe Seite 482) bzw. über die SET-Maske festgelegt werden. Die Eingabe im Feld SORT OPTION ist temporär und wird nicht gespeichert. Bezüglich der Sortierung der Dateienliste siehe auch Kommando SORT auf Seite 282.

Selektionsmaske : DOCUMENTATION



```
dd.mm.yy      hh:mm:ss        HOST: ........    USER-ID: ........    TSN: ....

         C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -   No File-Selection:  NO   -   Help:  ? / ?all / ??

Version x.xx   dd.mm.yy
```

Dieses Maskenfeld dient nicht zur Auswahl von Dateien oder Bibliothekselementen. Es kann hier der Name einer Datei angegeben werden, in die die am Bildschirm angezeigte Dateien- oder Bibliotheksliste in druckaufbereiteter Form geschrieben wird.

datei — Die am Bildschirm erscheinende Dateienliste soll in einer druckaufbereiteten Datei mit dem angegebenen Namen festgehalten werden. Die erzeugte Datei kann mit SPACE=E ausgedruckt werden.

datei, **E** — Eine bereits bestehende Dokumentationsdatei wird fortgeschrieben (Open=Extend).

datei, **INF** — Dateibeschreibungen, die mit dem Action-Code IM in einem CFS-Datenkatalog aufgenommen wurden (siehe Seite **421** und folgende (CFS-Dateiinformationssystem)), werden in die Dokumentationsdatei übernommen. Es werden nur die ersten 52 Bytes ungleich Blank in die Liste übernommen. Auf diese Art und Weise wird der Platz in jeder Druckzeile bis Spalte 132 mit dem Beginn der Dokumentation für die entsprechende Datei gefüllt.

datei, **INF, ALL [, SKIP]**

Wie oben, jedoch wird hier die vollständige Dateibeschreibung in die Liste übernommen.

SKIP — Der Zusatz bewirkt, daß Leerzeilen im Dokumentationstext bei der Ausgabe in die Datei unterdrückt werden.

datei, **INF, FORM= [*f* | NO] [, SKIP]**

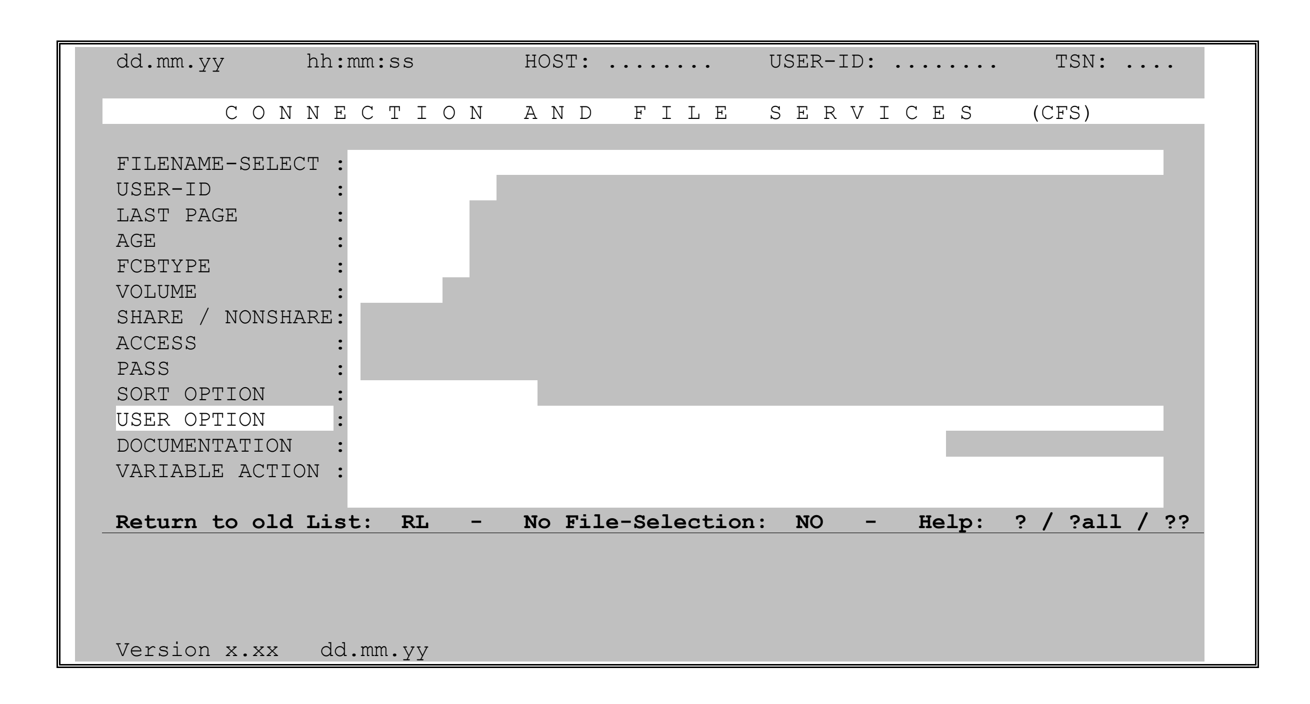
Es werden nur diejenigen Seiten aus der Dokumentation übernommen, die mit dem Format *f* (ein- bis dreistellige alphanumerische Bezeichnung, siehe Seite **424**), bzw. die ohne Format erfaßt wurden (FORM=NO).

Hinweise:

Wird in diesem Feld nichts eingetragen, so wird die Dateienliste standardmäßig in eine temporäre Datei mit dem Namen '#CFS.SAVELIST.[*jobname*|*tsn*]' (*tsn* nur bei fehlendem Jobnamen) geschrieben. Das Prefix '#CFS.SAVELIST.' kann vom Systemverwalter anders vorbesetzt werden (siehe CFSMAIN).

Zur Bedeutung der Dokumentationsdatei für den schnellen Wechsel zwischen zwei Dateienlisten (RL-Funktion) bzw. für eine Auswahl nach sehr differenzierten Selektionskriterien in mehreren Schritten (OLDLIST) wird auf die RL- und OLDLIST-Option im Feld FILENAME-SELECT (Seite **58**) verwiesen. Außerdem: Kommandos SL/RL (Save List/Restore List) Seite **275**/**281**, CFS-Parameter ADOC (Automatic Documentation) Seite **468** und DOC Seite **473**.

Selektionsmaske : USER OPTION



Allgemeine Hinweise:

Die im folgenden aufgeführten User Options bewirken, daß zusätzlich zu den von CFS standardmäßig angezeigten Dateimerkmalen noch ein oder mehrere, frei wählbare Merkmale in der Dateienliste ausgegeben werden. Durch Angabe einer Auswahlbedingung kann nach diesen Merkmalen auch selektiert werden. Mehrere User Options sind durch Semikolons zu trennen. Es können bis zu fünf User Options angegeben werden.

Der Wert für die als erstes angegebenen User Option wird in einer eigenen User Option Spalte in der Dateienliste eingetragen. Die Werte für eine evtl. angegebene zweite, dritte und vierte User Option überschreiben in der Dateienliste die Spalten PASS, AGE und LASTP (SPACE). Die fünfte User Option kann lediglich zur Selektion verwendet werden, da ihr Wert in der Dateienliste nicht dargestellt wird.

Der Benutzer kann eine oder mehrere User Options als Standard User Options vordefinieren. Ausführliche Informationen hierzu siehe Seite 483.

Beispiele:

`ALLO2;ALCNT;EXTNT;ALLOC`
Hiermit wird die User Option Spalte mit der Größe der Sekundärzuweisung (ALLO2) gefüllt. Die Spalte PASS wird mit der Anzahl der erfolgten Sekundärzuweisungen (ALCNT), die Spalte AGE mit der Anzahl der Extents (EXTNT) und die Spalte LASTP mit dem der Datei zugewiesenen Platz (ALLOC) gefüllt.

`ALCNT;EXTNT;;ALLOC`
Hiermit wird die User Option Spalte mit der Anzahl der erfolgten Sekundärzuweisungen (ALCNT) und die Spalte PASS mit der Anzahl der Extents (EXTNT) gefüllt. Die Spalte AGE wird nicht überschrieben (;;). Die Spalte LASTP wird mit dem der Datei zugewiesenen Platz (ALLOC) gefüllt.

`;;;VSN`
Diese Angabe bewirkt, daß in der ersten Spalte der Dateienliste (LASTP) die VSN der jeweiligen Datei angezeigt wird. Die Spalten AGE und PASS bleiben unverändert.

Zwei aufeinanderfolgende Semikolons ";;" bewirken, daß die entsprechende Spalte der Dateienliste nicht mit einer User Option überschrieben wird. Siehe hierzu die zwei letzten Beispiele.

Bei den Selektionsparametern kann anstelle von =*x* auch der einfache Wert *x* angegeben werden. Das Zeichen = wird in diesem Fall von CFS ergänzt.

ACCNT

Access-Counter.
Der Access-Counter wird bei jedem Lese- und Schreibzugriff auf die Datei erhöht.

Selektionsparameter: >*x* | <*x* | =*x*

ACL

Access-Control Liste.
Diese User Option zeigt an, ob für die Datei eine Full Access-Control Liste (FACL) definiert ist.

Selektionsparameter: Y | N

ADMINFO

Administrator Information (ab OSD V3.0).
Diese User Option zeigt den Inhalt der Administrator Information für die Datei an. Mit der Variablen Action ONXSET ADMINFO (siehe Seite **Fehler! Textmarke nicht definiert.**) bzw. mit dem Parameter ADM-INFORMATION im Kommando /MODIFY-FILE-ATTR kann der Systemverwalter einen bis zu acht Byte langen Text zu der Datei als Administrator Information hinterlegen.

Selektionsparameter: Y | N | '*info*'

Y | N — Administrator Information vorhanden ? Yes/No

'*info*' — Auswahl aller Dateien, deren Administrator Information den angegebenen Wert enthält.

ALCNT

Secondary Allocation Counter.
Der Secondary Allocation Counter zeigt an, wie oft eine Sekundärzuweisung von Speicherplatz für die Datei vorgenommen wurde.

Selektionsparameter: >*x* | <*x* | =*x*

ALLO2

Secondary Allocation.
Anzahl der PAM-Seiten für die Sekundärzuweisung. ALLO2 ist immer ein Vielfaches von 3.

Selektionsparameter: >*x* | <*x* | =*x*

ALLOC

Allocated Space.
Es wird der gesamte, einer Datei zugewiesene Speicherplatz angezeigt. Die erste Spalte der Dateienliste gibt standardmäßig den tatsächlich belegten Speicherplatz aus (Last Page). Zur Anzeige des zugewiesenen, jedoch nicht belegten Speicherplatzes siehe User Option FREE Seite **88**.

Selektionsparameter: >*x* | <*x* | =*x*

ATTR

Für Elemente einer PLAM-Bibliothek: Dateiattribute.
Diese User Option zeigt den Fcbtyp und die Recform-Eigenschaft, die das Element der PLAM-Bibliothek als BS2000-Datei besitzen würde.

Selektionsparameter: = S | I | P | V | F | U | [-]SV | [-]PAMK
S: Sam | I: Isam | P: Pam | V: Recform=V | F: Recform=F | U: Recform=U | SV: Sam und Recform=V | -SV: nicht Sam oder nicht Recform=V
PAMK: Blkctrl=Pamkey | -PAMK: Blkctrl=Data oder No

Hinweis:

Mehr Datei-Informationen zu einem bestimmten Bibliothekselement als in der User Option ATTR angegeben, erhalten Sie durch Eintragen des Action-Codes F bei dem entsprechenden Element (siehe Seite **193**).

Beispiele:

`ATTR F`	alle in der Bibliothek gespeicherten Dateien mit Recform=F.
`ATTR I`	alle in der Bibliothek gespeicherten Isam-Dateien.
`ATTR -SV`	alle in der Bibliothek gespeicherten Dateien, die entweder nicht vom Fcbtyp Sam sind oder nicht die Eigenschaft Recform=V besitzen.

AUDIT

Audit-Status für Datei.
Gibt die im Katalog verzeichnete Information zur Überwachung der Dateizugriffe durch System-Exit Routinen.

Selektionsparameter: NONE | SUCC | FAIL | ALL

AVAIL

Ausfallsicherheit (Verfügbarkeit) der Datei (ab OSD V3.0).
Die User Option zeigt an, ob für die Datei besondere Anforderungen bezüglich Ausfallsicherheit bestehen. Dateien mit erhöhter Ausfallsicherheit dürfen vom System nur auf geeigneten Datenträgern (z.B. DRV-Platten) abgelegt werden.
Mit der Variablen Action ONXSET AVAIL (siehe Seite **Fehler! Textmarke nicht definiert.**), dem Action-Code AV[H|S] bzw. mit dem Parameter AVAILABILITY im Kommando /MODIFY-FILE-ATTR kann der Verfügbarkeitsstatus von Dateien verändert werden.

STD: Die Datei hat keine besonderen Anforderungen bezüglich Ausfallsicherheit.
HIGH: Die Datei besitzt eine erhöhte Ausfallsicherheit.

Selektionsparameter: H[IGH] | S[TD]

BACL[F]

Basic Access Control Liste für Datei/Jobvariable/PLAM-Bibliothekselement.
Hinweis: Datenobjekte mit einer Basic Access Control Liste werden in der Dateienliste stets mit dem Zeichen '%' vor dem Fcbtyp gekennzeichnet.
Diese User Option zeigt an, ob und wenn ja, in welcher Weise für eine Datei/Jobvariable/PLAM-Bibliothekselement eine Access Control Liste definiert wurde. Mehr hierzu auf Seite **182**. Über die Access Control Liste kann die Zugriffsberechtigung für

das Datenobjekt wesentlich feiner abgestuft werden als dies mit der globalen Attribute Share=YES|NO, Access=WRITE|READ möglich ist. Der Zugriff kann eingeschränkt werden auf das Schreiben (W), Lesen (R) oder auf das Ausführen (X) von Programmen und Prozeduren. Für jede der genannten Zugriffsarten kann der Kreis der Benutzerkennungen, die auf das Datenobjekt Zugriff haben, eingeschränkt werden: keine Benutzerkennung (-), die Benutzerkennung des Eigentümers (Owners) der Datei (U), eine definierte Gruppe von Benutzerkennungen (G) oder auf alle Benutzerkennungen außerhalb der eigenen Gruppe (O). Es sind hierbei beliebige Kombinationen möglich.

Ausgabeformat der User Option: Für jede der drei Zugriffsarten REA, WRI und EXE wird jeweils eine dreistellige Beschreibung der Zugriffsdomänen ausgegeben. UGO besagt z.B., daß User (Owner), Group und Others Zugriff auf das Datenobjekt haben. U-- bedeutet, daß der Eigentümer (User), jedoch nicht die Gruppe und Others auf das Datenobjekt zugreifen können. Die erste dreistellige Buchstabenkombination steht für den lesenden Zugriff (REA), die mittlere Gruppe für den schreibenden Zugriff (WRI) und die rechte Gruppe für den ausführenden Zugriff (EXE). Beispiel für das Ausgabeformat:

```
REA WRI EXE
UG- U-- UGO
```

F

FSTAT-kompatibles Ausgabeformat. Anstelle des oben beschriebenen Formats werden die Access Control Eigenschaften in einem FSTAT-ähnlichen Format angezeigt. Die Darstellung erfolgt hier nicht nach Zugriffsarten gegliedert (s.o.), sondern nach den drei Zugriffsdomänen USER (OWNER), GROUP und OTHERS. Die Zugriffsrechte jeder Domäne werden in einer jeweils dreistelligen Buchstabenkombination dargestellt. RWX besagt z.B., daß ein lesender, schreibender und ausführender Zugriff auf das Datenobjekt möglich ist. R-X besagt, daß der lesende und ausführende Zugriff erlaubt ist. Beispiel für das Ausgabeformat:

```
USR GRP OTH
RWX R-X ---
```

Selektionsparameter:

Y: Es werden alle Dateien/Jobvariablen/Elemente einer PLAM-Bibliothek selektiert, für die eine Basic Access Control Liste festgelegt wurde.

N: Es werden alle Dateien/Jobvariablen/Elemente einer PLAM-Bibliothek selektiert, für die keine Basic Access Control Liste festgelegt wurde.

[W ><= *x*] [, R ><= *y*] [, X ><= *z*]

W = Schreibender, R = Lesender, X = Execute (Ausführender) Zugriff.
Die Zugriffsdomänen N, U, G, O sind bezüglich Größer (>) und Kleiner (<) folgendermaßen geordnet: N < U < G < O, O > U > G > N. Beispiel: siehe unten.

x, y, z: N | U | G | O (None / User-Id / User-Id Gruppe / alle Benutzer)

Es kann nach einer oder mehreren Zugriffsarten gleichzeitig selektiert werden. Bei Verwendung von '=' kann mehr als eine Domäne U/G/O direkt hintereinander angegeben werden.

[U = *a*] [, G = *b*] [, O = *c*]

U = User, G = Group, O = Others
a, b, c: R | W | X (Read / Write / Exec). Es können mehrere Zugriffsarten R/W/X direkt hintereinander angegeben werden.

Beispiele:

```
BACL W>U,R>U
```

Es werden alle Dateien selektiert, für die der schreibende und lesende Zugriff einem größeren Benutzerkreis als dem Eigentümer der Datei (User) gestattet ist. Abwei-

chend von der Darstellung der BACL-Eigenschaften im FSTAT-Kommando werden in der Dateienliste die Kurzbezeichnungen der Domänen ausgegeben, denen ein READ-, WRITE- und EXEC-Zugriff auf die Datei gestattet ist.

`BACLF G=RWX`
Es werden alle Elemente einer PLAM-Bibliothek selektiert, für die die Benutzergruppe einen lesenden, schreibenden und ausführenden Zugriff besitzt. BACLF: Das Darstellungsformat der BACL-Eigenschaften ist analog zu der des FSTAT-Kommandos, d.h. es werden zu jeder der Domänen USER, GROUP und OTHERS die entsprechenden Zugriffsrechte ausgegeben.

Hinweis:

Mit dem Action-Code AM (Access Modify) können die in der Liste dargestellten Basic Access Control Attribute durch Überschreiben direkt geändert werden. Für weitere Informationen siehe Seite **183**.

BACKUP

Backup-Level für Datei.
Selektionsparameter: A | B | C | D | E oder >*x* | <*x* | =*x*
(*x*: A | B | C | D | E)

BLKSIZE

Block Size.
Es wird ein Wert zwischen 1 und 16 ausgegeben. 2 bedeutet z.B., daß die Datei mit (STD,2) geblockt ist. Bei Banddateien, die nicht im Standardformat geblockt sind, wird der absolute Wert der Blocksize ausgegeben.

Selektionsparameter: >*x* | <*x* | =*x*

BLKCNT

Block Count.
Diese User Option ist nur für Banddateien sinnvoll. Es wird die Anzahl der von der Datei auf dem Band belegten Datenblöcke ausgegeben.

Selektionsparameter: >*x* | <*x* | =*x*

BLKCTRL

Block Control.
Diese User Option gibt an, ob und in welcher Form Pam-Keys als Bestandteil der Datei gespeichert sind.

PAMK	Datei enthält Pam-Keys, die in eigens dafür vorgesehenen Bereichen auf der Magnetplatte gespeichert sind.
DATA	SAM-Datei: Pam-Keys sind in den Datenblöcken untergebracht.
DATA2	ISAM-Datei mit Blockgröße 2048 Bytes. Pam-Keys sind in den Datenblöcken untergebracht.
DATA4	ISAM-Datei mit Blockgröße 4096 Bytes. Pam-Keys sind in den Datenblöcken untergebracht.
NO	Datei enthält keine Pam-Keys.

Selektionsparameter: =PAMKEY | =DATA[2|4] | =NO

CATBL

of Catalog-Block. Gibt die Nummer des PAM-Blocks im TSOSCAT, in dem der Katalogeintrag für die entsprechende Datei/Jobvariable gespeichert ist.
Mit der User Option CATBL kann der Systemverwalter leicht feststellen, ob die Katalogeinträge für die Dateien einer Kennung über viele Erweiterungen des TSOSCAT verstreut oder in einem zusammenhängenden Bereich gespeichert sind.

Selektionsparameter: >*x* | <*x* | =*x*

Beispiel:

`CATBL>4711`
Gibt alle Dateien einer Kennung, deren Katalogeinträge in einem Block >4711 des TSOSCAT abgelegt sind.

CCS

Name des Zeichensatzes, mit dem die Datei/das PLAM-Element bearbeitet wird (ab BS2000 V11.0). Mit den Selektionsangaben Y | N können alle Dateien ausgewählt werden, die mit einen Coded Character Set bearbeitet werden bzw. für die kein Coded Character Set definiert ist.
Ein Coded-Character-Set Namen kann für Dateien oder PLAM-Bibliothekselemente mit der Variablen Action ONXSET CCS=*name* zugewiesen werden.

Selektionsparameter: Y | N | [-]*name*

CFID

Coded File ID.
Es wird die kataloginterne Nummer der Datei als vier Byte lange sedezimale Zahl angezeigt. Jede Datei besitzt eine eindeutige CFID.

Selektionsparameter: >*x* | <*x* | =*x*
x steht für eine 8 stellige Sedezimalzahl oder für eine Angabe der Form: X'...'

CREA

Für Elemente einer PLAM-Bibliothek und Dateien ab BS2000 V11: Zeit in Tagen, die seit der Erstellung der Datei bzw. der erstmaligen Aufnahme des Elements in die PLAM-Bibliothek vergangen sind. Dieses Datum ist nicht identisch mit dem Datum der letzten Änderung.

Selektionsparameter: >*x* | <*x* | =*x* bzw.
in Datumsform: >|<|= *dd.* [*mm.* [*yy*]]
Falls *mm.yy* oder *yy* weggelassen werden, so werden die aktuellen Monats- und Jahresangaben ergänzt.

Hinweise:

Durch die Reorganisation der Bibliothek geht das CREA-Date der Elemente verloren. Alle Elemente enthalten als CREA das Datum der Reorganisation.

Mit dem Kommando DATE (Seite 473) wird veranlaßt, daß das Change Date nicht als Zeitangabe in Tagen, sondern in Form eines Datums angezeigt wird.

Beispiele:

`CREA 0`
Es werden alle Dateien selektiert, die heute neu eingerichtet wurden.

`CREA<7`
Es werden alle Elemente einer PLAM-Bibliothek selektiert, die vor weniger als 7 Tagen erstmalig aufgenommen wurden.

`CREA<15.05.`
Es werden alle Elemente selektiert, die nach dem 15.05. des aktuellen Jahres erstmalig in die Bibliothek aufgenommen wurden.

DELDATE

Lösch-Freigabedatum für Datei (ab OSD V3.0).
Es wird die Anzahl von Tagen bzw. das Datum angezeigt, ab dem die Datei ohne Berücksichtigung der Schutzattribute gelöscht werden darf. Die Datei kann vor diesem Datum jederzeit modifiziert und unter Berücksichtigung der Schutzattribute auch gelöscht werden.
Mit der Variablen Action ONXSET DELDATE (siehe Seite **Fehler! Textmarke nicht definiert.**), dem Action-Code DD bzw. mit dem Parameter FREE-FOR-DELETION im Kommando /MODIFY-FILE-ATTR kann dieses Datum verändert werden.

Selektionsparameter: >*x* | <*x* | =*x* bzw.

in Datumsform: >|<|= *dd.* [*mm.* [*yy*]]

Falls *mm.yy* oder *yy* weggelassen werden, so werden die aktuellen Monats- und Jahresangaben ergänzt.

Hinweise:

Mit dem Kommando DATE (Seite **473**) wird veranlaßt, daß die Retention-Period nicht als Zeitangabe in Tagen, sondern als Datum in der Form YMMDD angezeigt wird. Y steht hierbei für die letzte Stelle des Jahres (z.B. 6 für 1996).

DELTA[1]

Delta-Speicherung in PLAM-Bibliotheken.
Delta-Speicherung bedeutet, daß verschiedene Versionen des gleichen Elements in einer Form abgelegt sind, bei der jeweils nur die geänderten Sätze bezüglich der Vorgängerversion gespeichert werden. Durch die Delta-Technik kann im Vergleich zur Vollspeicherung der Elemente eine bedeutende Reduzierung des Speicherplatzes erreicht werden.

Die User Option DELTA erfüllt die folgenden Funktionen:

a) Bei deltagespeicherten PLAM-Elementen werden alle vorhandenen Versionen mit einem eigenen Eintrag in der Dateienliste dargestellt.
Standard: Bei delta-gespeicherten Bibliothekselementen wird jeweils nur die höchste vorhandene Version in der Dateienliste ausgegeben.

b) In der User Option Spalte wird bei jeder Element-Version die für die Deltaspeicherung verwendete Vorgängerversion (Basis) angezeigt. Die erste Version eines Elements hat sich selbst als Vorgänger und wird durch einen Stern (*) gekennzeichnet.
DELTA: Die Basisversion eines jeden Elements wird in Form des internen Versionszählers ausgegeben (analog LMS bei der Einstellung: PAR TOC=D).
DELTA1: Es werden die letzten vier Stellen der echten Versionsbezeichnung des Vorgänger-Elements ausgegeben.

c) Wird eine in der Dateienliste enthaltene delta-gespeicherte Elementversion mit ONXADD in eine andere PLAM-Bibliothek übertragen, in der ein Element dieses Namens noch nicht existiert, so wird das Element in deltagespeicherter Form in die neue Bibliothek übernommen. Im Normalfall, d.h. ohne User Option DELTA wird bei ONXADD ein delta-gespeichertes Element als Vollelement übertragen, falls ein Element dieses Namens in der Zielbibliothek nicht existiert.

Selektionsparameter: Y | H | N

Y Es werden nur die delta-gespeicherten Elemente der PLAM-Bibliothek selektiert. In der Dateienliste werden alle Elementversionen angezeigt.

H Es werden nur die delta-gespeicherten Elemente der PLAM-Bibliothek selektiert. In der Dateienliste wird nur die höchste (aktuellste) Version eines jeden Delta-Elements angezeigt.

N Es werden alle nicht delta-gespeicherten Elemente der Bibliothek selektiert.

DI[SK]WR Disk Write (ab BS2000 V11.0)
Diese User Option gibt an, ob sich die Datei zur Bearbeitung im DAB Store-in-Cache eignet.

C Die Datei ist für Store-in-Cache geeignet. Die Daten der Datei müssen erst nach dem CLOSE in einem konsistenten Zustand auf der Platte sein (BS2000 Default für temporäre Dateien).

I Die Datei ist nicht für Store-in-Cache geeignet. Die Daten der Datei müssen auch nach einem Systemabsturz in einem konsistenten Zustand auf der Platte sein (BS2000 Default für permanente Dateien).

Selektionsparameter: =C | =I

EXTNT Anzahl Extents.
Extent ist ein zusammenhängender Speicherbereich auf einer Magnetplatte, in dem die Datei bzw. ein Teil der Datei abgelegt ist.

Selektionsparameter: >*x* | <*x* | =*x*

DEVICE Device-Typ (Plattentyp), auf dem die Dateien gespeichert sind.
Diese User Option zeigt den Device-Typ der Platten- bzw. Banddateien an. Bei Angabe eines festen Device-Typs als Selektionsbedingung werden die VSN's der Dateien mit diesem Device-Typ, angezeigt.

Selektionsparameter: =*device*
Für *device* kann ein beliebiger Platten- bzw. Bandtyp, wie er im FSTAT-Kommando ausgewiesen wird, angegeben werden.

Zeichenfolgen in den Daten (Namen) von Dateien/Bibliotheks-elementen/Jobvariablen suchen

FIND [*n*,] *param* [=W *datei* [, E|O]] [, N] [, NO] [, C|NC] [, F] [, SKIPF] [, TYPE= *t1*, *t2*,...]

Auswahl aller Datenobjekte, die einen oder mehrere der angegebenen Suchbegriffe in ihren Daten bzw. in ihrem Namen enthalten. Die gefundenen Treffersätze und die entsprechenden Dateinamen können wahlweise in eine druckaufbereitete List-Datei geschrieben werden (=W *datei*). In der Dateienliste wird die Anzahl der Sätze ausgegeben, in denen der Suchbegriff mindestens einmal vorkommt.

n Beschränkung der Suche auf die ersten *n* Sätze. (Standard: alle Sätze).

param einfache oder mehrfache Suchanweisung.

einfache Suchanweisung: [*col*] [*p*] *item*

col — Spaltenbereich in dem die gesuchte Zeichenfolge beginnen muß.

:col1-col2: Das erste Zeichen der gesuchten Zeichenfolge muß im Spaltenbereich zwischen col1 und col2 beginnen.

:col1: Die Zeichenfolge wird nur an der angegebenen Spalte col1 gesucht und muß dort beginnen.

>:col1: | <:col1: Die Zeichenfolge wird im Bereich ab Spalte col1 bis Satzende (>) bzw. vom Satzanfang bis Spalte col1 gesucht (<)

Standard: Die Suche erstreckt sich von Spalte 1 eines jeden Satzes bis zum jeweiligen Satzende.

p

> Suche nach einer Zeichenfolge > *item*
< Suche nach einer Zeichenfolge < *item*
- Suche nach einer Zeichenfolge ungleich *item*

Standard: Suche nach einer Zeichenfolge = *item*

item — Suchzeichenfolge: C'*string*' | L'*string*' | X'*string*' | A'*string*'
C'*string*' kann zu '*string*' abgekürzt werden.
L'*string*': Kleinbuchstaben in *string* werden nicht in Großbuchstaben umgesetzt.
A'*string*': Bei der Suche werden Klein- und Großbuchstaben gleich behandelt. Beispiel: Der Suchbegriff A'CFS' wird als Treffer erkannt, falls im Datensatz die Zeichenfolge 'CFS' oder 'cfs' oder eine beliebige Kombination aus Groß- und Kleinbuchstaben enthalten ist.

Enthält *string* Hochkommas ('), so müssen diese verdoppelt angegeben werden ('').

Mehrfachsuche - Format 1: *param* [*vk param*] [*vk param*] .....

param — einfaches Suchargument gemäß der oben beschriebenen Syntax.

vk — Verknüpfungsoperator mit dem vorausgegangenen einfachen Suchargument.

, — Oder-Verknüpfung.

+ — Und-Verknüpfung.

*****[*n*] — Wildcard-Verknüpfung: Und-Verknüpfung, jedoch muß das zweite Suchitem im Datensatz nach dem ersten Suchitem vorkommen. Der optionale Zusatz *n* legt die Anzahl der zwischen den beiden Suchitems zu stehenden Trennzeichen fest.

Es können beliebig viele Konstrukte der Art *vk such* aneinandergereiht werden.
Bei Verknüpfung durch die Und-Bedingung '+' können auch geklammerte Ausdrücke angegeben werden: (.. , .. , ...)+(.. , .. , ...). Ein Beispiel hierzu ist auf Seite **315** beschrieben.

Hinweise:

Die Und-/Oder-Verknüpfung ist jeweils auf einen Datensatz bezogen. Dies bedeutet, daß beide Suchbegriffe im selben Satz enthalten sein müssen.

Eine ausführlichere Darstellung der Verknüpfungsoperationen finden Sie auf Seite **313**.

Mehrfachsuche - Format 2: ( )

Durch () wird eine spezielle Such-Maske angefordert. In ihr stehen dem Benutzer 18 Bildschirmzeilen zur Aufnahme von umfangreichen Suchargumenten zur Verfügung. Für eine Beschreibung der Such-Maske siehe Seite **315**.

Mehrfachsuche - Format 3: %*name* | %? [, *jrsave*]

%*name* ist die Bezeichnung eines Parameter-Sets, unter dem zu einem früheren Zeitpunkt ein komplexes Suchargument gespeichert wurde (SP %*name*). Näheres hierzu siehe Seite **315**.
%?: Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Markieren mit 'x' kann ein Parameterset ausgewählt werden. Durch Markieren mit 'm' (Modify) wird der Inhalt des Parametersets angezeigt und kann noch verändert werden.

jrsave ist der Name der Datei, in der der Parameter-Set gespeichert ist. Standard: CFS.JRSAVE [.*jobname*] .

Mehrfachsuche - Format 4: (*s-dat*)

s-dat ist der Name einer Datei (SAM), in der die Suchargumente festgehalten sind. Näheres hierzu siehe Seite **316**.

Mehrfachsuche - Format 5: ? | ?,%*name*

Durch ? wird das zuletzt definierte Suchargument wieder angezeigt.
Durch ?,%*name* wird die mit der in %*name* enthaltenen mehrfachen Suchbedingung ausgefüllte Such-Maske (siehe Format 2) angezeigt.

Wegschreiben der Treffersätze

=W *datei* [, E | O]

Die Treffersätze werden in eine druckaufbereitete Ausgabedatei geschrieben. Die Namen der entsprechenden Datenobjekte werden in der Write-Datei ebenfalls dokumentiert. Bei Angabe der NO-Option (siehe unten) werden nur die Namen dokumentiert.

E — Die Write-Datei wird mit Open=Extend eröffnet.

O — Die Write-Datei wird mit Open=Output eröffnet.

Der Zusatz =W *datei* [, E | O] kann auch in der Such-Maske (siehe oben: Mehrfachsuche - Format 2) bzw. in *s-dat* (siehe oben: Mehrfachsuche - Format 4) im Anschluß an den letzten Suchstring angegeben werden.

FIND ..., N — Es werden die Namen der gerade durchsuchten Datenobjekte am Bildschirm angezeigt.

FIND ...., NO — Names Only. Die Stringsuche erstreckt sich nur auf die Namen der Datenobjekte. Die Namen der Datenobjekte werden in die Write-Datei geschrieben. Der Anwendungsbereich dieser Option liegt vor allem im Durchsuchen mehrerer Bibliotheken nach Elementen mit einem bestimmten Namen. Siehe hierzu auch das Beispiel 2 weiter unten.

FIND ..., NC — No Check for Continuation. Die Prüfung auf Fortsetzung eines BS2000-Kommandos /einer Makro-/Quellprogramm-Anweisung wird unterdrückt.

FIND ..., C — Es werden bei Prozeduren/Makros/Quellprogrammen alle Sätze als logische Einheit betrachtet, die durch Fortsetzungszeichen miteinander verbunden sind. Dies bedeutet, daß z.B. der zusammengesetzte Suchbegriff 'FCB'*'BLKCTRL=DATA' in der Datei als Treffer gefunden wird, obwohl der Teilstring 'FCB' in einer Zeile, und der Teilstring 'BLKCTRL=DATA' in einer Folgezeile enthalten ist.
Bei der Ausgabe der Treffersätze in eine Datei werden alle Sätze der logischen Einheit in die Write-Datei übertragen. Falls der Suchstring vollständig in einem der Teilsätze enthalten ist, so wird dieser Teilsatz in der Ausgabedatei durch das Zeichen 'X' in Spalte 2 hervorgehoben.

FIND ..., F — Find First. Nach dem ersten gefundenen Treffer wird die Suche für das Datenobjekt beendet. Es wird dann ein Treffer gemeldet. Mit dieser Option läßt sich auf schnelle Weise feststellen, ob in einem Datenobjekt ein vorgegebener Suchbegriff enthalten ist oder nicht. Standardmäßig werden nach einem gefundenen Treffer auch alle weiteren Sätze des Datenobjekts nach dem Suchbegriff durchsucht. Die Anzahl der Treffer wird in diesem Fall mit dem korrekten Wert angezeigt.

FIND ..., SKIPF
Skip Found. Es werden die Datenobjekte in der Liste aufgeführt, die den in der Suchbedingung angegebene String in keinem Satz enthalten.

FIND ..., TYPE=*t1*,*t2*, ...
Dieser Parameter ist nur von Bedeutung, falls Dateien durchsucht werden, hinter denen sich PLAM-Bibliotheken verbergen.
Die Suche erstreckt sich nur auf Elemente in den Bibliotheken, die vom angegebenen Typ sind. Beispiel: TYPE=J oder TYPE=J,S
Standardmäßig werden alle Elemente mit beliebigen Typ-Bezeichnungen in der Bibliothek durchsucht.

Hinweise:

Im Zusammenhang mit der Write-Option (=W ..) wird auf das Kommando **REWR** (Zurückschreiben der Write-Datei in die ursprünglichen Datenobjekte) hingewiesen. Das Rewrite-Kommando REWR (siehe Seite 271) stellt das Gegenstück zur Write-Option dar.

Bei Auswahl von Dateien sind die in der Trefferliste aufgeführten Dateien i.a. nicht in alphabetisch aufsteigender Reihenfolge sortiert.

Wurde hinter dem Namen der Write-Datei keine der Optionen E/O (Extend/ Overwrite) angegeben, so gilt folgende Regelung:

Falls die Write-Datei im aktuellen CFS-Lauf zum ersten Mal angesprochen wird, so wird als Open-Modus in jedem Fall O (Overwrite) angenommen, d.h. die Datei wird neu angelegt bzw. überschrieben.

Falls in mehreren Selektionsläufen nacheinander die gleiche Write-Datei angegeben wurde, so wird sie standardmäßig mit Open=Extend eröffnet.

Durch die E-/O-Option kann vom Benutzer ein vom Standardfall abweichender Open-Modus angegeben werden.

CFS durchsucht PAM-Dateien, die Bibliotheken repräsentieren auf der Ebene der Bibliothekselemente und nicht auf der elementaren Datensatz- oder Blockebene. Falls dieses Verhalten nicht gewünscht ist, kann z.B. der Parameter TYPE=Z angegeben werden.

Aufgrund des oben beschriebenen standardmäßigen Verhaltens der User Option FIND wird die Gesamtanzahl der Treffersätze über alle Elemente in einer Bibliothek bzw. die Anzahl der Namen der Bibliothekselemente, die einen der gesuchten Strings enthalten.

Bei der User Option FIND werden Banddateien sowie durch HSMS ausgelagerte (migrierte) Dateien nicht berücksichtigt.

Beispiele:

```
FIND 'FLAG'*'STATEMENTS'=W DAT.FLAGS
```
In der Trefferliste werden alle Dateien/Bibliothekselemente ausgewählt, die mindestens einen Datensatz enthalten, in dem die Zeichenfolge 'FLAG' und weiter hinten die Zeichenfolge 'STATEMENTS' enthalten ist. Die gefundenen Treffersätze werden in die SAM-Datei DAT.FLAGS geschrieben.

```
FIND 'A1015'=W L.FNDLIST,NO
```
Anforderung: Es ist ein Quellprogramm mit dem Namen A1015 in mehreren Bibliotheken der eigenen Kennung bzw. im gesamten BS2000-System zu suchen. Die in Frage kommenden Bibliotheken können im aktuellen Fall durch die Eingabe von LIB im Feld FILENAME-SELECT vorselektiert werden. Die oben angegebene User Option FIND durchsucht die Namen aller Elemente der vorausgewählten Bibliotheken nach dem Vorkommen des Strings A1015. In der Datei L.FNDLIST werden die Elementnamen, die einen Treffer gebracht haben, zusammen mit dem entsprechenden Bibliotheksnamen aufgeführt. In der Dateienliste werden nur die Bibliotheken aufgeführt, die mindestens ein Element enthalten, dessen Name die angegebene Suchbedingung erfüllt. Mit dem Action-Code NP bzw. NPF kann sich der Benutzer das Inhaltsverzeichnis jeder einzelnen Bibliothek anzeigen lassen.

```
FIND :1:'$T'
```
Der Inhalt der Jobvariablen muß in Spalte 1 mit '$T' beginnen (Monitor-Jobvariablen, die auf normal beendete Aufträge hinweisen).

Weitere Beispiele für Suchargumente und die Verknüpfung von mehreren Suchargumenten finden Sie auf Seite **314** - **316**.

FREE[%][ISAM][,PLAM] Anzahl der freien PAM-Seiten einer Datei, d.h. zugewiesener, jedoch nicht belegter Bereich. Free = Allocated Space - Last Page

% Prozentanteil der freien PAM-Seiten bezüglich der Gesamtgröße der Datei.

ISAM Handelt es sich bei einer Datei um eine ISAM-Datei, so wird unter Zuhilfenahme des Kommandos SHOW-FREE-ISAM-SPACE die Anzahl der unbenutzten Datenblöcke ausgegeben. Die Option ISAM kann nur unter der Benutzerkennung TSOS angegeben werden. Das FSC-Softwareprodukt SPACEOPT V2.0 muß außerdem installiert sein.

PLAM Handelt es sich bei einer Datei um eine PLAM-Bibliothek, so wird der innerhalb der Bibliothek nicht benutzte Platz zu der Anzahl der freien Seiten hinzugezählt. Die Option PLAM kann ab LMS Version 3.0 benutzt werden.

Selektionsparameter: >*x* | <*x* | =*x*

GUARD Guard-Schutz für Dateien/PLAM-Bibliothekselemente (ab BS2000 V11.0).
Hinweis: Datenobjekte mit einem Guardschutz werden in der Dateienliste mit dem Zeichen '$' vor dem Fcbtyp gekennzeichnet.

R Der lesende Zugriff (Read) ist mit Guard geschützt
Das Zeichen N an der Stelle von R besagt, daß kein Guard für den Lesezugriff vergeben wurde. Das Datenobjekt kann daher nicht gelesen werden.

W Der schreibende Zugriff (Write) ist mit Guard geschützt
Das Zeichen N an der Stelle von W besagt, daß kein Guard für den Schreibzugriff vergeben wurde. Das Datenobjekt kann daher nicht modifiziert werden.

X Der ausführende Zugriff (Exec) ist mit Guard geschützt
Das Zeichen N an der Stelle von X besagt, daß kein Guard für den ausführenden Zugriff vergeben wurde. Das Datenobjekt kann daher nicht ausgeführt werden.

Mit den Selektionsangaben Y | N werden alle Dateien ausgewählt, für die ein Guard-Schutz bzw. kein Guard-Schutz aktiviert ist.

Mit den Selektionsangaben [-] R|W|E [=*name*] werden alle Dateien ausgewählt, für die eine bestimmte Zugriffsart [nicht] mit einem beliebigen Guard bzw. mit einem Guard des angegebenen Namens geschützt ist. Der Pseudo-Guardname 'NONE' ist anzugeben, um Datenobjekte auszuwählen, die mit Guard geschützt sind, jedoch für die gewünschte Zugriffsart kein Guard definiert ist. Für diese Datenobjekte ist kein Zugriff der entsprechenden Art (R/W/X) möglich.

Selektionsparameter: Y | N | [-]R|W|E [=*name*]

Beispiele:

`GUARD W`
Es werden alle Dateien/Bibliothekselemente selektiert, die bezüglich des schreibenden Zugriffs mit einem Guard geschützt sind.

`GUARD R=$TSOS.READGUARD`
Es werden alle Dateien/Bibliothekselemente selektiert, die bezüglich des lesenden Zugriffs mit einem Guard des Namens $TSOS.READGUARD geschützt sind.

`GUARD X=NONE`
Es werden alle Dateien/Bibliothekselemente selektiert, die mit Guard geschützt sind und für die kein Guard bezüglich des ausführenden Zugriffs definiert ist. Diese Datenobjekte können nicht ausgeführt werden.

HOLD

Hold-Status für PLAM-Elemente. Ab LMS Version 3.0 können Elemente einer Bibliothek ausgeliehen werden. Dies bedeutet, daß die Benutzerkennung des Entleihers im Inhaltsverzeichnis der Bibliothek vermerkt wird. Nur Benutzer, die unter der Kennung des Entleihers angeloggt sind, können das Element in der gleichen oder in einer höheren Version ändern oder kopieren.

Die User Option HOLD zeigt den Ausleihstatus der einzelnen Elemente. Die Anzeige 'N' bedeutet, daß das Element nicht ausgeliehen ist. Falls das Element ausgeliehen ist, wird die Benutzerkennung des Entleihers angezeigt.

Selektionsparameter: N | Y [:*user-id*] (*user-id* ohne $ und Punkt)

Durch die Selektionsangabe Y werden alle ausgeliehenen Elemente ausgewählt.
Mit Y:*user-id* werden alle Elemente ausgewählt, die unter einer bestimmten Kennung ausgeliehen wurden.

Hinweise:

Bibliothekselemente können mit dem Action-Code HOLD ausgeliehen werden. Mit NHOLD kann der Ausleihstatus zurückgesetzt werden. Für weitere Informationen hierzu siehe Seite **195**.

Für eine ausführliche Beschreibung der Auswirkungen von HOLD wird auf das LMS-Benutzerhandbuch Version 3.0 verwiesen: Kapitel 3 (Elemente), Abschnitt 3.6 (Elementschutz/Datenschutz), Thema: "Übersicht über notwendige Rechte für LMS-Aktionen".

HSMS

HSMS Storage-Level. Es wird der Migrationsstatus der Datei angezeigt.

Dateien, die in einem bestimmten Zeitraum nicht durch Lese- oder Schreiboperationen angesprochen wurden, werden von HSMS (Hierarchical Storage Management System) in den Hintergrundspeicher 1 ausgelagert (migriert). Werden die Dateien auch hier für eine gewisse Zeitspanne nicht angesprochen, so erfolgt die Auslagerung auf den Hintergrundspeicher 2. Der Hintergrundspeicher 1 kann z.B. aus langsamen Magnetplatten mit großer Kapazität bestehen. Der Hintergrundspeicher 2 wird normalerweise auf Magnetbandkassetten realisiert.

S0 Die Datei ist im Public-Bereich (Vordergrundspeicher) abgelegt.
S2 Die Datei wurde auf den Hintergrundspeicher 2 ausgelagert (migriert).

Selektionsparameter: S*n* bzw. >S*n* | <S*n* (*n* = 0, 2)

Hinweise:

Migrierte Dateien sind in der Dateienliste auch durch das Zeichen # in der Spalte LASTP zu erkennen.

Zum Thema HSMS siehe auch Seite **92** (User Option MIGRATE) bzw. **200** (Action-Codes MA/ML/MI).

INF

Zeigt an, ob für die Datei/Jobvariable bzw. für das Bibliothekselement eine Beschreibung im CFS-Informationskatalog vorliegt und ggf. welches Format für die Erfassung verwendet wurde. (Siehe Action Codes I/IM Seite **196** bzw. Seite **421** und folgende (CFS-Datei-Informationssystem)).

Selektionsparameter: [Y | N | S] [, FORM=*xxx*] [, '*string*']

Y | N Inf-Text vorhanden ? Yes/No

S Wie Y. Zusätzlich werden die ersten 15 Byte des Informationstextes in der Dateienliste angezeigt.

FORM=*xxx* Auswahl bezüglich des bei der Erfassung verwendeten Formats.

'*string*' Auswahl aller Datenobjekte, die in ihrem Dokumentationstext den angegebenen Begriff (Suchstring) enthalten.

Hinweise:

Bei der Angabe des Parameters FORM= bzw. bei der Auswahl nach einem Suchbegriff wird der Selektionsparameter Y intern ergänzt und muß daher vom Benutzer nicht angegeben werden.

Es kann mehr als ein Selektionsparameter angegeben werden. In diesem Fall sind die einzelnen Parameter durch Kommas zu trennen.
`FORM=SX0,'UTM'` selektiert z.B. alle Datenobjekte, für die ein Dokumentationstext über die benutzerdefinierte Maske SX0 eingegeben wurde. Im Dokumentationstext muß außerdem der Begriff 'UTM' enthalten sein. Klein-/Großschreibung ist ohne Bedeutung.

INDEX Zeigt die Anzahl der eingerichteten Indizes einer ISAM-Datei an. Für K-ISAM Dateien und NK-ISAM Dateien ohne Sekundärindizes wird der Wert 1 ausgegeben, da diese Dateien nur eine Indexstruktur besitzen. Für NK-ISAM Dateien mit n Sekundärindizes (BS2000-Kommando CREate-Alternate-Index) wird der Wert n+1 ausgegeben.

Selektionsparameter: >*x* | <*x* | =*x*

Hinweise:

Mit der Selektionsangabe >1 werden alle NK-ISAM Dateien ausgewählt, für die mindestens ein Sekundärindex eingerichtet wurde.

Für eine vollständige Information über Namen und Keypos/Keylen aller eingerichteten Sekundärindizes einer Datei kann der Action-Code SI (Show Alternate Index, siehe Seite 204) angegeben werden.
Zur Anzeige einer NK-ISAM Datei gemäß der Sortierreihenfolge eines Sekundärschlüssels kann das Kommando DA (Data Access, siehe Seite 302) verwendet werden.

IOP[ERF] Disk-IO Performance (ab BS2000 V11.0)
Diese User Option gibt an, ob die Datei für die IO-Verarbeitung in Verbindung mit einem Cache geeignet ist.

STD: Die Datei soll nicht über einen Cache bearbeitet werden.
HIGH: Die Datei ist für Cache-Verarbeitung gut geeignet und soll, wenn möglich, über einen Cache bearbeitet werden.
VHIGH: Die Datei ist für Cache-Verarbeitung sehr gut geeignet. Wenn möglich, werden alle Seiten im Globalspeicher (GS) gehalten.

Selektionsparameter: ><= STD | HIGH | VHIGH
Die Reihenfolge der Werte ist: STD < HIGH < VHIGH

IOU[SAGE] Disk-IO Usage (ab BS2000 V11.0)
Diese User Option gibt an, auf welche Operationen sich das Performance-Attribut IOP der Datei (IO-Performance, siehe oben) bezieht.

RW: Das Performance-Attribut bezieht sich auf Lese- und Schreiboperationen.
W: Das Performance-Attribut bezieht sich nur auf Schreiboperationen.
R: Das Performance-Attribut bezieht sich nur auf Leseoperationen.

Selektionsparameter: =RW | =W | =R

KEYLEN Länge des Schlüssels bei ISAM-Dateien.

Selektionsparameter: >*x* | <*x* | =*x*

Beispiel:

`KEYLEN>8`
Es werden alle ISAM-Dateien ausgewählt, die einen Schlüssel mit einer Länge größer als 8 Byte besitzen.

KEYPOS Position des Schlüssels in ISAM-Dateien.

Selektionsparameter: >*x* | <*x* | =*x*

Beispiele:

`KEYPOS>5`
Es werden alle ISAM-Dateien ausgewählt, bei denen der Schlüssel am Anfang des Satzes steht.

`KEYPOS;KEYLEN>0`
Es werden alle ISAM-Dateien ausgewählt, Als Eigenschaften werden die Werte für Keypos und Keylen angezeigt.

LACC[A] Last Access Date. Zeit in Tagen, die seit dem letzten Lese- oder Schreibzugriff auf die Datei bzw. das Bibliothekselement vergangen ist.

A Der Parameter A bewirkt, daß ab OSD V1.0 in der Dateien- bzw. PLAM-Elementliste die Felder AGE und TIME mit dem Last Access Date und der Last Access Time gefüllt werden. Das Alter des Datenobjekts wird in der User Option Spalte angezeigt.

Selektionsparameter: >*x* | <*x* | =*x* bzw.
in Datumsform: >|<|= *dd.* [*mm.* [*yy*]]
Falls *mm.yy* oder *yy* weggelassen werden, so werden die aktuellen Monats- und Jahresangaben ergänzt.

Hinweise:

Um die User Option LACC für PLAM-Bibliothekselemente zu nutzen, muß LMS in einer Version >= 3.0 installiert sein. Die Aktualisierung des Last Access Date muß für die Bibliothek eingeschaltet werden. Dies geschieht, indem der Action-Code LAKL bei der Bibliothek selbst oder bei einem beliebigen Element der Bibliothek eingetragen wird. Im Vorschaltmodul CFSMAIN kann der Systemverwalter einen Indikator setzen, der bewirkt, daß neue Bibliotheken von CFS automatisch mit der Last Access Date Option angelegt werden.

Mit dem Kommando DATE (Seite 473) wird veranlaßt, daß das Last Access Date nicht als Zeitangabe in Tagen, sondern als Datum in der Form YMMDD angezeigt wird. Y steht hierbei für die letzte Stelle des Jahres (z.B. 6 für 1996).

Beispiele:

`LACC=0`
Es werden alle Dateien ausgewählt, auf welche heute lesend oder schreibend zugegriffen wurde.

`LACC>7`
Es werden alle Dateien ausgewählt, auf welche vor mehr als 7 Tagen zum letzten Mal lesend oder schreibend zugegriffen wurde.

`LACC<15.05.`
Es werden alle Dateien ausgewählt, auf die nach dem 15.05. des aktuellen Jahres zuletzt zugegriffen wurde.

MANCLASS Die User Option gibt an, ob die HSMS-Funktionen Dateisicherung, Archivierung und Langzeitarchivierung über eine mit HSMS definierte Management-Klasse gesteuert werden. Die User Option steht ab ab OSD V3.0 zur Verfügung.
Mit der Variablen Action ONXSET MANCLASS (siehe Seite **Fehler! Textmarke nicht definiert.**) bzw. mit dem Parameter MANAGEMENT-CLASS im Kommando /MODIFY-FILE-ATTR kann die Management-Klasse von SM-Dateien verändert werden.

Selektionsparameter: *class* | N[ONE]

MAXRL Länge des längsten Datensatzes in Datei/ Bibliothekselement/ Länge des Datenteils der Jobvariablen.

Diese User Option bewirkt, daß alle Datenobjekte sequentiell vom ersten bis zum letzten Satz durchgelesen werden, um den längsten Datensatz zu bestimmen.
Eine Ausnahme bilden Magnetbänder. Hier kann die maximale Satzlänge nicht bestimmt werden.

Dateien mit Fcbtype=PAM werden bei der Selektion ausgelassen.

Selektionsparameter: >*x* | <*x* | =*x*

MIGRATE Auslagerungsfähigkeit (Migrierbarkeit) von Dateien durch das Hierarchical Storage Management System (HSMS).

A Migration Allowed. Die Datei kann uneingeschränkt ausgelagert werden.
I Migration Inhibited. Die Datei soll nicht durch HSMS ausgelagert werden. Eine kurzzeitige Auslagerung zu Reorganisationszwecken ist aber erlaubt.
F Migration Forbidden. Die Datei darf nicht ausgelagert werden.
Selektionsparameter: A | I

Hinweis: Zum Thema HSMS siehe auch Seite 89 (User Option HSMS) bzw. 200 (Action-Codes MA/ML/MI).

NO **Names Only**: Es werden nur die Namen der selektierten Datenobjekte in der Dateienliste angezeigt. Die übrigen Merkmale wie LASTP, TYP, AGE, usw. sind in der Dateienliste mit Spaces gefüllt.

NOF Diese User Option ist nur zusammen mit FILENAME-SELECT : OLDLIST *datei* sinnvoll. Die User Option bewirkt, daß alle in der OLDLIST-Datei enthaltenen Dateieinträge unverändert und ohne erneute Existenzprüfung für die Dateinamen in die aktuelle Dateienliste übernommen werden. Ebenso werden alle Dateiangaben, wie z.B. LASTP, AGE unverändert in die Dateienliste übernommen.

Hinweise:

Durch diese User Option lassen sich bei großen Benutzerkennungen, bei der Selektion über alle Dateien im System, sowie bei großen PLAM-Bibliotheken Verkürzungen in der Suchzeit erreichen.

Zum Thema Verkürzung der Suchzeiten bei PLAM-Bibliotheken siehe auch Hinweis auf Seite 66.

Bei Verwendung der User Option NO (Names-Only) dürfen nur in den Selektionsfeldern FILENAME-SELECT und USER-ID Auswahlbedingungen eingetragen werden. Im Feld FCBTYPE darf nur J (Jobvariable) bzw. ein Elementtyp bei PLAM-Bibliotheken angegeben werden. Wird in einem anderen Feld eine Selektionsbedingung eingetragen, so wird die User Option NO ignoriert.

Im Zusammenhang mit der Names-Only User Option wird auf den Action-Code U (Update File-List) hingewiesen. Mit diesem Action-Code werden nachträglich die Dateiattribute in die Liste eingetragen.

NOC No Cat-Id. Bei der Auswahl von Dateien mit dem Schlüsselwort FSTAT im Selektionsfeld FILENAME-SELECT, werden die standardmäßig vorangestellten CAT-IDs aus den Dateinamen entfernt.

Diese Option kann notwendig werden beim Sichern von Dateien mit ARCHIVE. Aufgrund der Archive-Anweisung: PARAM CAT-ID=NO dürfen in den Namen der zu sichernden Dateien keine CAT-IDs vorkommen.

OPEN Diese User Option zeigt alle Dateien, die von Tasks der eigenen Kennung eröffnet sind. Voraussetzung für die User Option OPEN ist, daß Sie unter TSOS angeloggt sind oder daß der Systemverwalter unter TSOS den CFS-Holdertask gestartet hat (/ENTER CFS.S.LMSLIB(CFSHT)).

Selektionsparameter: [**S**,] [**I**|**O**|**RL**|**E**|**N**|**PR**|**PL**|**SEC**[**EX**]|**ALL**] [,**TSN=***tsn*]
[,**POOL=** *isam-pool*] [,**LINK** [=*link*]] [,**D-IO**[**S**]|**A-IO**[**S**]
[><=*x*] [,INT=*sec*] [,RW]]

- S neben Benutzertasks werden auch die von Systemtasks eröffneten Dateien berücksichtigt. Die S-Option alleine ist gleichbedeutend mit IOEN,S. Durch OPEN S werden somit alle aktiven und inaktiven TFT-Einträge angezeigt.
- I alle mit Input eröffneten Dateien
- O alle mit Output, Outin, Inout, Update, Extend eröffneten Dateien
- RL wie O, es werden jedoch nur die mit SHARUPD=YES eröffneten ISAM-Dateien ausgewählt, für die eine Satz- oder Blocksperre besteht. In der Spalte Open-Modus wird dies mit #IO angezeigt zusammen mit der entsprechenden TSN.
- E alle mit Exec eröffneten Dateien (geladene Programme, SDF-SyntaxDateien)
- N alle mit File zugewiesenen, jedoch noch nicht bzw. nicht mehr eröffneten Dateien (inaktive TFT-Einträge)
- PR alle durch ein Print-Kommando gesperrten Dateien (Open-Typ PR),
  alle durch ein TRANSFER-FILE gesperrten Dateien (Open-Typ FT),
  alle durch einen System-Lock gesperrten Dateien (open-Typ SY).
- PL alle geöffneten PLAM-Bibliotheken, sowie die Namen der darin geöffneten Elemente. Zusätzlich werden auch TSN und Open-Modus (I/O/U/E) angezeigt.
  Der Action-Code D (Display) auf geöffnete PLAM-Elemente ist nicht möglich.

SEC   alle mit einem SECURE-Kommando shareable/exclusiv reservierten Dateien
SECEX alle mit einem SECURE-Kommando exclusiv reservierten Dateien
SECEX ist eine Teilmenge von SEC (s.o.)
ALL   Abkürzung für IOPRPLSEC, falls der Parameter TSN= nicht angegeben wurde.
Abkürzung für IOENPRPLSEC, falls der Parameter TSN= angegeben wurde.

Standard:

IOPRSEC, falls der Parameter TSN= nicht angegeben wurde.
IOENPRSEC, falls der Parameter TSN= angegeben wurde.

Die Selektionsbedingungen I O E N PR PL SEC[EX] ALL dürfen in beliebigen Kombinationen angegeben werden (z.B. IO, OIE, IOENSEC, ALLE).

TSN=*tsn*

Anzeige aller TFT-Einträge zu der angegebenen TSN. Für tsn kann auch die Bezeichnung einer Systemtask angegeben werden. Beispiel: TSN=M.

POOL=*isam-pool*

Durch Angabe eines Pool-Namens werden alle NK-ISAM Dateien ausgewählt, die an diesem Pool angeschlossen sind.

LINK Es werden die zum Eröffnen der Dateien verwendeten Linknamen angezeigt. Standard: Es werden keine Linknamen ausgegeben.

link Es werden nur die mit dem angegebenen Linknamen eröffneten Dateien selektiert. Beispiel: LINK=ECERDLOD selektiert alle geladenen Programme.

D-IO [><=*x*]

Delta IO. In der AGE-Spalte der Dateienliste wird die <u>Anzahl der IO's</u> angezeigt, die in einem bestimmten Zeitintervall (Standard=30 Sekunden) für jede einzelne Datei angefallen sind. Im Unterschied zu der Option D-IOS (siehe unten) werden Dateien, die von mehreren Tasks eröffnet sind, in der Liste mehrfach aufgeführt zusammen mit der entsprechenden TSN und der Anzahl der IO's. Durch ><=*x* werden alle Dateien ausgewählt, die mehr als, weniger als oder genau *x* IO's verbraucht haben. Beispiel: `OPENS D-IO>0`

D-IOS [><=*x*]

Delta IO Summe. In der AGE-Spalte der Dateienliste wird die <u>Anzahl der IO's</u> angezeigt, die in einem bestimmten Zeitintervall (Standard=30 Sekunden) für jede einzelne Datei angefallen sind. Für Dateien, die von mehreren Tasks eröffnet sind, wird der Summenwert der IO's über alle Tasks ausgegeben. In der Spalte TSN wird dann '****' eingetragen, da sich die Anzahl der IO's nicht auf eine Task bezieht. Durch ><=*x* werden alle Dateien ausgewählt, die mehr als, weniger als oder genau *x* IO's verbraucht haben.

Beispiel:
User-Option: `OPENS,D-IOS>0` und Sort-Option: `AGE,D`
Die eröffneten Dateien werden nach der Anzahl der verbrauchten IO's absteigend sortiert.

INT=*sec*

Länge des Zeitintervalls für D-IO in Sekunden. Standard: INT=30.
Um die IO-Werte ausgeben zu können, wird CFS für die angegebene Anzahl von Sekunden in einen Wartezustand versetzt.

RW In den beiden ersten Anzeigespalten der Dateienliste wird für jede Datei anstelle von LASTP und TYP die Anzahl der verbrauchten Read- und Write-IO's angezeigt.

Ohne Angabe von Selektionsbedingungen werden alle im Input- oder Output-Modus eröffneten Dateien angezeigt. Nicht ausgegeben werden Dateien, die

- im EXEC-Modus eröffnet sind (geladene Programme, SDF-Syntaxdateien)
- Dateien, die mit einem File-Kommando zugewiesen, jedoch nicht eröffnet sind

A-IO [><=x]

Accumulated IO. In der AGE-Spalte der Dateienliste wird die Anzahl der IO's angezeigt, die seit dem Eröffnen der Datei durch die entsprechende TSN angefallen sind. Im Unterschied zu der Option A-IOS (siehe unten) werden Dateien, die von mehreren Tasks eröffnet sind, in der Liste mehrfach aufgeführt. Durch ><=x werden alle Dateien ausgewählt, die mehr als, weniger als oder genau x IO's verbraucht haben. Beispiel: `OPENS A-IO>100`
Die Option A-IO und A-IOS (siehe unten) steht erst ab OSD V3.0 zur Verfügung.

A-IOS [><=x]

Accumulated IO Summe. In der AGE-Spalte der Dateienliste wird die Anzahl der IO's angezeigt, die seit dem Eröffnen der Datei durch die entsprechende TSN angefallen sind. Für Dateien, die von mehreren Tasks eröffnet sind, wird der Summenwert der IO's über alle Tasks ausgegeben. In der Spalte TSN wird dann '****' eingetragen, da sich die Anzahl der IO's nicht auf eine Task bezieht. Durch ><=x werden alle Dateien ausgewählt, die mehr als, weniger als oder genau x IO's verbraucht haben.

Ausgaben der User Option OPEN:

In der Spalte User Option der Dateienliste wird die TSN des Prozeßes angezeigt, der die Datei eröffnet bzw. der ein File-Kommando für die Datei abgesetzt hat. Daneben wird der Modus angezeigt, in dem die Datei eröffnet ist:

I Input
O Output
IO Inout
OI Outin
U Update
E Extend
EX Exec
R Reverse
⊔ Inaktiver TFT-Eintrag. Datei ist gegenwärtig nicht eröffnet
S Datei ist mit einem Secure-Kommando shareable reserviert
SE Datei ist mit einem Secure-Kommando exclusiv reserviert
PR Datei wird mit PRINT ...,LOCK=YES ausgedruckt

Ein Stern (*) vor dem Open-Modus besagt: Die Datei ist mit SHARUPD=YES eröffnet.

Das Zeichen (#) vor dem Open-Modus IO besagt: Die Datei ist mit SHARUPD=YES eröffnet und von der links stehenden TSN wird eine Satz-/Blocksperre gehalten.

Bei geöffneten NK-ISAM Dateien wird in der ersten Spalte der Dateienliste (LASTP) der Name des verwendeten ISAM-Pools angezeigt.

Hinweise:

Falls keine SORT OPTION angegeben wurde, werden die durch die User Option OPEN selektierten Dateien in der Reihenfolge der erzeugenden TSNs und hierbei in

der Reihenfolge der abgesetzten FILE-Kommandos aufgelistet. Eine Datei kann mehrmals in der Liste vorkommen, falls sie von verschiedenen Tasks eröffnet wird.

Die Anzeige aller TSN's, die eine bestimmte Datei in der Dateienliste eröffnet haben, ist mit dem Action-Code OPN möglich. Der Action-Code OPN zeigt im Gegensatz zur User Option OPEN auch die TSN's an, die nicht unter der eigenen Benutzerkennung angeloggt sind.

Nach der Selektion von Dateien mit der User Option OPEN kann mit dem Action-Code S ein Kommando /STA *tsn* für die ausgewiesene TSN angefordert werden.

Der momentane Inhalt von SAM- und PAM-Dateien, die im Output-Modus eröffnet sind, kann mit dem Action-Code sichtbar gemacht werden.

IO-Auswertung mit der Option D-IO: Während der Dauer der Auswertung kann das Programm KDCMON nicht verwendet werden. Umgekehrt kann während einer KDCMON-Messung die Option D-IO nicht verwendet werden. Eine D-IO Auswertung kann auch nicht von mehreren Benutzern gleichzeitig aufgerufen werden. In diesem Fall wird die Meldung "KDCMON active" ausgegeben. Sollte diese Meldung ausgegeben werden, wobei das Programm KDCMON aber nachweislich nicht geladen ist, so kann die Lock-Situation beseitigt werden, indem anstelle von D-IO, die Option X-IO verwendet wird.

Eine mit D-IO erzeugte Dateienliste kann mit dem CFS-Kommando SORT D-IO,D nach der Anzahl der verbrauchten IO's absteigend sortiert werden. Die gleiche Sortierung kann auch über das Feld SORT OPTION der Selektionsmaske erreicht werden. Hier ist als Sortierkriterium jedoch AGE,D anzugeben.

Beispiele:

`OPEN TSN=9011`
Alle zum Zeitpunkt der Selektion existierenden TFT-Einträge der TSN 9011.

`OPEN TSN=9011,D-IO>0,INT=60`
Alle Dateien der TSN 9011, die in den letzten 60 Sekunden IO's verbraucht haben. In der Spalte D-IO der Dateienliste wird die Anzahl der IO's angezeigt. Vor Ausgabe der Dateienliste wird CFS für 60 Sekunden in einen Wartezustand versetzt.

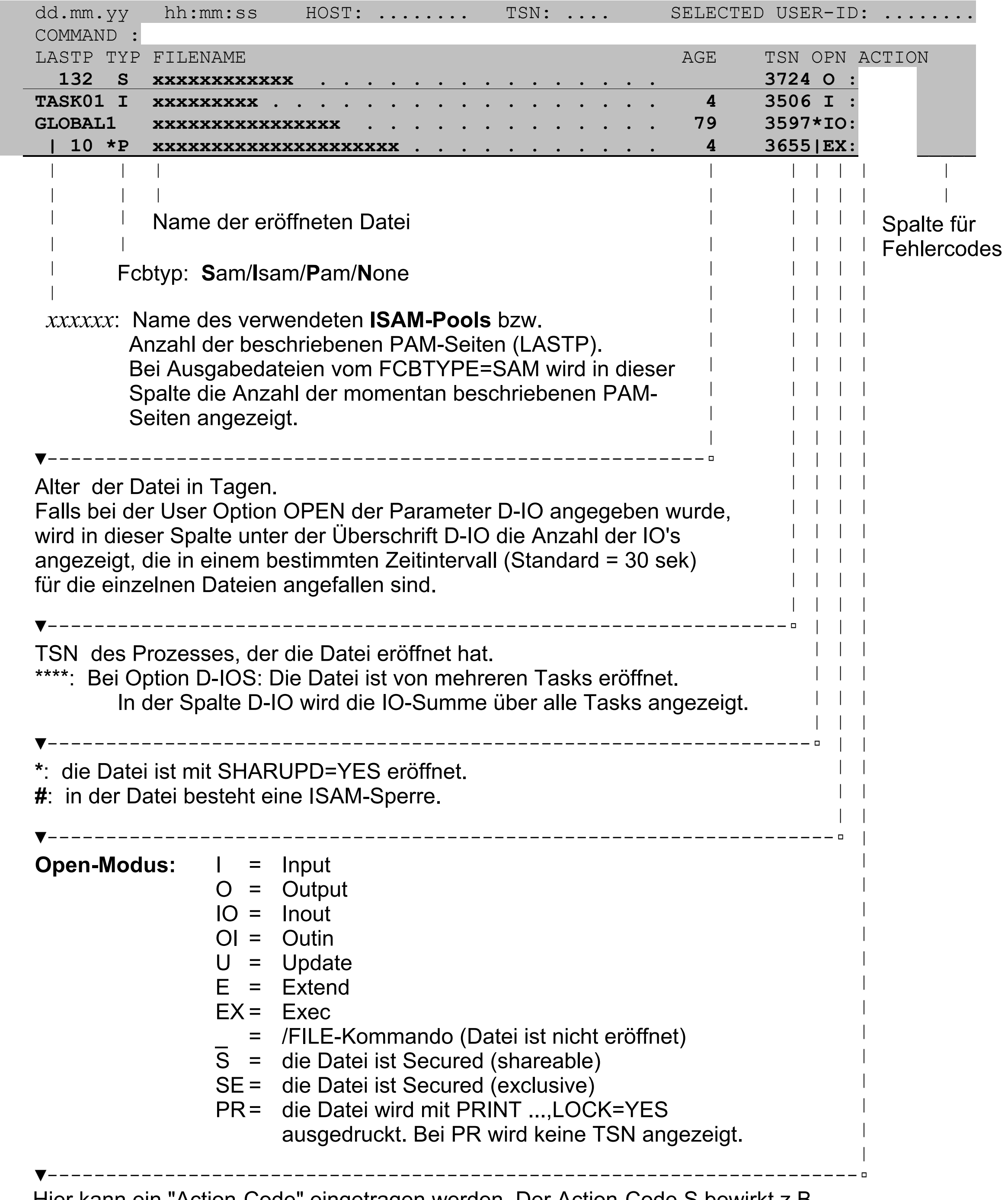
`OPEN O`
Alle im Output-Modus eröffnete Dateien von Tasks der eigenen Kennung.

`OPEN PR`
Alle durch ein Print-Kommando mit Parameter LOCK=YES gesperrten Dateien.

`OPEN IOEN,LINK`
Alle aktiven bzw. nicht aktiven TFT-Einträge mit den zugehörigen Linknamen.

`OPEN POOL=$TASK01`
Alle eröffneten NK-ISAM Dateien, die an dem Pool $TASK01 angeschlossen sind.

Format der Dateienliste für User Option OPEN:



Hier kann ein "Action-Code" eingetragen werden. Der Action-Code S bewirkt z.B. die Ausgabe der Status-Informationen zur angezeigten TSN. Der Action-Code D (Display) kann auch auf SAM-Dateien angewendet werden, die im Output-Modus eröffnet sind, wie z.B. Ablaufprotokolle.

Auswahl aller aktiven Monitor Jobvariablen

ONEVT Es werden alle Monitor Jobvariablen der eigenen Benutzerkennung angezeigt, die zum aktuellen Zeitpunkt mit einem ONEVT-Kommando bzw. Makro überwacht werden. Die User Option ONEVT kann in eingeschränktem Umfang auch für Nicht-TSOS Benutzer bereitgestellt werden, falls der Systemverwalter unter TSOS einen CFS-Holdertask startet (/ENTER CFS.S.LMSLIB(CFSHT)).

Selektionsparameter: keine

Ausgaben der User Option ONEVT:

In der Spalte User Option der Dateienliste werden die TSN's der Prozesse angezeigt, die die Jobvariable mit einem ONEVT-Kommando überwachen. Falls mehrere Tasks eine Jobvariable überwachen, wird der JV-Name entsprechend oft in der Dateienliste aufgeführt.

Hinweise:

Falls keine SORT OPTION angegeben wurde, werden die durch die User Option ONEVT selektierten Jobvariablen in der Reihenfolge der erzeugenden TSNs aufgelistet. Eine Jobvariable kann mehrmals in der Liste vorkommen, falls sie von verschiedenen Tasks überwacht wird.

Die Auswahl von Jobvariablen über die User Option ONEVT kann mit allen anderen Bedingungen, insbesondere mit der Auswahl über FILENAME-SELECT kombiniert werden.

Nach der Selektion von Jobvariablen mit der User Option ONEVT kann in der Dateienliste mit dem Action-Code S ein Status Kommando der ausgewiesenen TSN angefordert werden.

Offene Dateien bei ARCHIVE mit sichern ?

OPNBACK Für jede Datei wird angezeigt, ob sie in geöffnetem Zustand (Open=Output/Update/Extend/Inout/Outin) bei ARCHIVE-Sicherungen mit gesichert wird (Voraussetzung: PARAM OLS=YES ist angegeben). Die Sicherungsmöglichkeit einer Output-geöffneten Datei wurde für UDS-Dateien geschaffen und ist im Systemkatalog im Byte IDCEDMS2 (Indikator IDCEFAR) dauerhaft vermerkt.
In CFS kann dieser Indikator mit der Variablen Action ONXOPNBACK [=Y|N] für beliebige Benutzerdateien gesetzt oder zurückgenommen werden. Für weitere Informationen zur OPNBACK-Funktion siehe Seite **Fehler! Textmarke nicht definiert.**.

Selektionsparameter: = Y | N

PASS Vorhandene Passworte für Datei/Jobvariable/Bibliothekselement.

Die User Option zeigt an, ob für ein Datenobjekt READ-/WRITE-/EXEC-Passworte definiert sind. Die User Option ermöglicht es, die BACL's und die vorhandenen Passworte in einer Liste zusammen darzustellen. Um dies zu erreichen, ist im Feld USER OPTION: BACL;;PASS einzutragen. Damit werden die Art der Passworte in der AGE-Spalte angezeigt.
Selektionsparameter: W | R | E | X | Y | N
Die Selektionsparameter können in gleicher Weise auch im Feld PASS der Selektionsmaske angegeben werden.

RECNT

Anzahl der Datensätze (Datenblöcke) in einer Datei/ einem Bibliothekselement.

Diese User Option hat zur Folge, daß alle Datenobjekte sequentiell vom ersten bis zum letzten Satz durchgelesen werden. Eine Ausnahme bilden PAM-Dateien. Hier kann die Anzahl der beschriebenen PAM-Blöcke aus der Lastpage-Angabe im Katalog bestimmt werden.

Selektionsparameter: >*x* | <*x* | =*x*

RECFORM

Satzformat der Datei bzw. des PLAM-Bibliothekselements.

V: variable Satzlänge
F: fixed, wobei die RECSIZE mit angezeigt wird
U: undefined

Selektionsparameter: =V|U|F [>*x*|<*x*|=*x*]
Mit der optionalen Angabe >|<|=*x* werden alle Dateien mit einer Recsize >/</= *x* ausgewählt.

RECSIZE

Satzlänge der Datei.

Es wird die Satzlänge der Datei gemäß dem Katalogeintrag angezeigt. Zusätzlich zur Recsize wird der Recform-Parameter mit ausgegeben (F/V).

Selektionsparameter: >*x* | <*x* | =*x*

Beispiel:

```
RECSIZE>0
```

Auswahl aller Recform=F Dateien sowie aller Recform=V Dateien mit einem Recsize-Wert > 0.

RELSP

Space Minus ignorieren ? (ab BS2000 V11.0)

- N Kommandoaufrufe zum Freigeben von nicht mehr benötigtem Speicherplatz (z.B. FILE ..,SPACE=-*nnn*) werden für die Datei ausgeführt.
- Y Das Freigeben von nicht benötigtem Speicherplatz ist aufgrund des Katalogeintrags für die Datei nicht möglich.

Selektionsparameter: =Y | =N

REPAIR

Datei in inkonsistentem Zustand ?

- Y Die Datei ist aufgrund eines Systemabsturzes in einem inkonsistenten Zustand und kann nicht geöffnet werden (z.B. von ARCHIVE). Um die Datei wieder herzustellen, muß das Kommando VERIFY ...,REPAIR=YES abgesetzt werden.
- N Die Datei nicht beschädigt.

Selektionsparameter: =Y | =N

RETPD

Retention Period. Zeit in Tagen bis zum Verfallsdatum der Datei.

Mit dem Kommando DATE (Seite **473**) wird veranlaßt, daß die Retention-Period nicht als Zeitangabe in Tagen, sondern als Datum in der Form YMMDD angezeigt wird. Y steht hierbei für die letzte Stelle des Jahres (z.B. 6 für 1996).

Selektionsparameter: >*x* | <*x* | =*x* bzw.
in Datumsform: >|<|= *dd.* [*mm.* [*yyyy*]]
Falls *mm.yyyy* oder *yyyy* weggelassen werden, so werden die aktuellen Monats- und Jahresangaben ergänzt.

Hinweise:

Falls die Date-Option (siehe oben) aktiviert ist und falls die Selektionsbedingung in der Form >|<|= *dd.mm.yyyy* angegeben wurde, werden in der Dateienliste auch bereits vergangene Retention-Period angaben in der Form YMMDD angezeigt. Es können z.B. mit <dd.mm.yyyy (heutiges Datum) alle Dateien mit einer Retention-Period in der Vergangenheit ausgewählt werden.

Die User Option RETPD zeigt bei Archive-Directory Einträgen das Verfallsdatum des Datenträgers (Magnetband/Kassette), auf dem die Datei gesichert ist.

Beispiele:

`RETPD=0`
Auswahl aller Dateien, die ein Verfallsdatum von heute haben, d.h. die überschrieben werden dürfen.

`RETPD>7`
Auswahl aller Dateien, die eine Retention-Period von mehr als 7 Tagen haben (gerechnet vom heutigen Tage an).

`RETPD>01.01.95`
Auswahl aller Dateien mit einem Verfallsdatum im Jahr 1995 oder später.

RT

Für Elemente einer PLAM-Bibliothek:
Selektiert bezüglich des Vorkommens bestimmter Satzarten in den Elementen einer PLAM-Bibliothek.

Selektionsparameter (obligatorisch): >*x* | <*x* | =*x* (0 <= *x* <= 255).

Beispiel:

`RT=26`
Auswahl aller Elemente der PLAM-Bibliothek, die **REP-Sätze** (Satzart 26) enthalten. Diese Satzart kommt normalerweise nur in Elementen vom Typ R (Relocatable) vor. Elemente vom Typ R können mit den Action-Codes D/M angeschaut und ggf. modifiziert werden.

S0MIGR

Die User Option zeigt an, ob die Datei innerhalb der Speicherebene S0 auf einen anderen Volume-Set des gleichen SMS-Pubsets verlagert werden darf. Die Verlagerung kann explizit mit HSMS oder implizit durch das BS2000 erfolgen.
Die User Option steht ab ab OSD V3.0 zur Verfügung.
Mit der Variablen Action ONXSET S0MIGR (siehe Seite **Fehler! Textmarke nicht definiert.**) bzw. mit dem Parameter S0-MIGRATION im Kommando /MODIFY-FILE-ATTR kann die S0-Migrierbarkeit von SM-Dateien verändert werden.
A Allowed. Eine Verlagerung der Datei ist möglich
F Forbidden. Eine Verlagerung der Datei ist nicht möglich

Selektionsparameter: A | F

SAVE

Ist die Datei aufgrund von Änderungen im Dateninhalt bei der nächsten ARCHIVE-Sicherung mitzusichern ? Falls die geänderte Datei bereits in einem voran-

gegangenen ARCHIVE-Lauf erfaßt wurde, so ist im Katalogeintrag der Datei Current Version # gleich der Backup Version #.

Selektionsparameter: Y | N (Y: Datei ist aufgrund aktueller Änderungen mit ARCHIVE zu sichern).

SAVE

Werden mit der Angabe DIR im Feld FILENAME-SELECT Einträge eines Archive-Directories selektiert, so bewirkt die User Option SAVE ohne Angabe von Parametern, daß alle Sicherungsversionen einer Datei/Jobvariablen in der Liste angezeigt werden und nicht nur die letzte (aktuellste) Version, wie dies standardmäßig der Fall ist.

Durch den Zusatz SAVE CNS werden zusätzlich auch alle CNS-Einträge ("Catalogued but not Saved") in der Liste angezeigt.

SDAT

Für Elemente einer PLAM-Bibliothek:
System-Datum des Elements umgerechnet in die Anzahl von Tagen bis heute.
5 bedeutet z.B., daß das Element vor 5 Tagen zuletzt geschrieben wurde.

Selektionsparameter: >*x* | <*x* | =*x* bzw.
in Datumsform: >|<|= *dd.* [*mm.* [*yy*]]
Falls *mm.yy* oder *yy* weggelassen werden, so werden die aktuellen Monats- und Jahresangaben ergänzt.

Hinweise:

Bei einer Reorganisation der Bibliothek wird das System-Datum aller Elemente auf das aktuelle Datum gesetzt. Das User-Datum der Elemente bleibt unverändert.

Das System-Datum wirkt bei PLAM-Elementen wie das Change Date bei Dateien, d.h. es wird bei jedem schreibenden Zugriff automatisch vom System aktualisiert. Das User-Datum kann hingegen auch vom Benutzer gesetzt und verändert werden (siehe Variable Action ONXSETUDAT).

Soll das System-Datum als AGE-Kriterium bei der Auswahl von PLAM-Elementen verwendet werden, so ist das Kommando SDAT (siehe Seite **481**) anzugeben.

Mit dem Kommando DATE (Seite **473**) wird veranlaßt, daß das User-Datum des Bibliothekselements nicht als Zeitangabe in Tagen, sondern als Datum in der Form YMMDD angezeigt wird. Y steht hierbei für die letzte Stelle des Jahres (z.B. 6 für 1996).

SNAM

Für Elemente einer PLAM-Bibliothek und PLAM-Bibliotheken in der Dateienliste:
Selektiert alle Elemente, für die im Inhaltsverzeichnis ein Verweis auf einen Sekundärnamen existiert.

Selektionsparameter:

Auf Dateiebene (PLAM-Bibliotheken):

>*x* | <*x* | =*x* Es werden alle Bibliotheken ausgewählt, die mehr als, weniger als oder genau *x* Elemente mit Sekundärnamen enthalten.

sname Alle Bibliotheken, die mindestens ein Element mit dem angegebenen Sekundärnamen enthalten.
In der User Option Spalte wird die Anzahl der Elemente mit dem gewünschten Sekundärnamen angezeigt.

such Alle Bibliotheken, die mindestens ein Element mit einem Sekundärnamen enthalten, der der angegebenen Suchbedingung '...' genügt.

such=W *datei* Alle Bibliotheken, die mindestens ein Element mit einem Sekundärnamen enthalten, der der angegebenen Suchbedingung genügt. Die gefundenen Sekundärnamen werden in die angegebene Datei geschrieben.

Auf Dateiebene ist die Angabe eines Selektionskriteriums zwingend notwendig.

Auf Bibliotheksebene (LMS *bibliothek*):

>*x* | <*x* | =*x* Es werden alle Elemente ausgewählt, die mehr als, weniger als oder genau *x* Sekundärnamen enthalten.

sname Alle Elemente, die den angegebenen Sekundärnamen enthalten.
In der User Option Spalte wird der Sekundärname angezeigt. In der Spalte ## wird die Gesamtanzahl der Sekundärnamen zu diesem Element angegeben.

such Alle Elemente, die einen Sekundärnamen enthalten, der der angegebenen Suchbedingung '...' entspricht.

such=W *datei* Alle Bibliotheken, die mindestens ein Element mit einem Sekundärnamen enthalten, der der angegebenen Suchbedingung genügt. Die gefundenen Sekundärnamen werden in die angegebene Datei geschrieben. Alle Sekundärnamen werden in die Datei geschrieben durch die Angabe SNAM ''=W *datei*

sname/attr bestimmter Sekundärname, bestimmtes Attribut
/attr beliebiger Sekundärname mit diesem Attribut
-*sname* bestimmter Sekundärname darf nicht vorkommen
-*/attr* bestimmtes Attribut darf nicht vorkommen

Falls auf Bibliotheksebene kein Selektionskriterium angegeben wurde, so werden nur die Elemente vom Typ L und R der Bibliothek ausgewählt. In der Spalte User Option wird die Gesamtanzahl der Sekundärnamen zusammen mit dem jeweils ersten Sekundärnamen jedes Elements angezeigt.

Hinweis:

Bei Elementen vom Typ R (Relocatable) sind stets Verweise auf Sekundärnamen gespeichert. Mit dem Action-Code F werden alle Sekundärnamen eines Elements am Bildschirm ausgegeben. Für nähere Informationen siehe Seite 193.

STOCLASS Die User Option zeigt an, ob für die Datei eine besondere Storage-Klasse definiert ist. Eine Storage-Klasse bestimmt den Ablageort der Datei innerhalb eines SM-Pubsets.
Die User Option steht ab ab OSD V3.0 zur Verfügung.
Mit der Variablen Action ONXSET STOCLASS (siehe Seite **Fehler! Textmarke nicht definiert.**) bzw. mit dem Parameter STORAGE-CLASS im Kommando /MODIFY-FILE-ATTR kann die Storage-Klasse von Dateien verändert werden.

Selektionsparameter: N | Y | *class*

TIME

Für Elemente einer PLAM-Bibliothek / Dateien / Jobvariable / Archive-Directory Einträge: Selektiert bezüglich der Uhrzeit der letzten Änderung.
Selektionsparameter (obligatorisch): >*x* | <*x* | =*x*
x: Zeitangabe in der Form *hh:mm* .
Wird *:mm* weggelassen, so wird :00 ergänzt.

TIMEX

Für Dateien: Es wird die Uhrzeit der letzten Änderung als sechsstellige Zahl, d.h. sekundengenau ausgegeben.

UDAT

Für Elemente einer PLAM-Bibliothek:
USER-Datum des Elements umgerechnet in die Anzahl von Tagen bis heute.
5 bedeutet z.B., daß das Element ein USER-Datum von 5 Tagen vor dem heutigen Datum besitzt.

Selektionsparameter: >*x* | <*x* | =*x* bzw.
in Datumsform: >|<|= *dd.* [*mm.* [*yy*]]
Falls *mm.yy* oder *yy* weggelassen werden, so werden die aktuellen Monats- und Jahresangaben ergänzt.

Hinweise:

Bei einer Reorganisation der Bibliothek bleibt das User-Datum der Elemente im Gegensatz zum System-Datum unverändert.

Soll statt des User-Datums das System-Datum als AGE-Kriterium bei der Auswahl von PLAM-Elementen verwendet werden, so ist das Kommando SDAT (siehe Seite **481**) anzugeben.

Mit dem Kommando DATE (Seite **473**) wird veranlaßt, daß das User-Datum des Bibliothekselements nicht als Zeitangabe in Tagen, sondern als Datum in der Form YMMDD angezeigt wird. Y steht hierbei für die letzte Stelle des Jahres (z.B. 6 für 1996).

USERINFO

Benutzerinformation (ab OSD V3.0).
Diese User Option zeigt den Inhalt der Benutzerinformation für die Datei an. Mit der Variablen Action ONXSET USERINFO (siehe Seite **Fehler! Textmarke nicht definiert.**) bzw. mit dem Parameter USER-INFORMATION im Kommando /MODIFY-FILE-ATTR kann ein bis zu acht Byte langer Text zu der Datei als Benutzerinformation hinterlegt werden.

Selektionsparameter: Y | N | '*info*'

Y | N — Benutzerinformation vorhanden ? Yes/No

'*info*' — Auswahl aller Dateien, deren Benutzerinformation den angegebenen Wert enthält.

VAR

Für Elemente einer PLAM-Bibliothek:
Variante des Bibliothekselements. Die Variante eines Bibliothekselements ist eine 4-stellige Zahl, die anfangs den Wert 1 besitzt und die bei jedem schreibenden Zugriff auf das Bibliothekselement um 1 erhöht wird.

Selektionsparameter: >*x* | <*x* | =*x*

VER Für Elemente einer LMS-ISAM/PLAM-Bibliothek erfüllt die User Option VER zwei Funktionen:

1) Es wird die Anzahl der unter dem Elementnamen gespeicherten Versionen ausgegeben.

2) Es wird nur die höchste Version jedes Elements in der Dateien-liste angezeigt. Dies gilt auch für Elemente einer LMS ISAM-Bibliothek und für nicht deltagespeicherte PLAM-Elemente. Bei deltagespeicherten PLAM-Elementen wird auch ohne User Option VER in der Dateienliste nur die höchste Version angezeigt.

Selektionsparameter: >*x* | <*x* | =*x*

Hinweis:

Die User Option VER ignoriert eine in der Selektionsmaske oder im NP-Kommando evtl. angegebene Sort Option. Die Dateienliste wird stets nach Namen aufsteigend sortiert.

VCNT Volume-Counter. Anzahl der Magnetplatten (Magnetbänder), auf denen die Datei Speicherplatz belegt hat.

Selektionsparameter: >*x* | <*x* | =*x*

VSN erste VSN, auf der die jeweilige Datei Speicherplatz belegt hat. Falls eine Datei auf mehr als einer VSN Speicherplatz belegt hat, so wird dies durch das Zeichen '+' rechts von der VSN bzw. bei Archive Directory Einträgen durch das Zeichen '*' links von der VSN angezeigt.
Die User Option VSN kann auch bei Archive-Directories (NPDIR *directory*) angegeben werden. In diesem Fall wird die VSN des ersten Bandvolumes angezeigt, auf dem die Datei gesichert wurde.

Selektionsparameter: *vsn*

VOLSET Die User Option zeigt die Bezeichnung des Volume-Sets, auf dem die Datei innerhalb eines SM-Pubsets gespeichert ist. Die User Option steht ab ab OSD V3.0 zur Verfügung.
Mit dem Parameter VOLUME-SET im Kommando /CREATE-FILE kann das Volume-Set von neu anzulegenden Dateien bestimmt werden.

Selektionsparameter: *catid* (z.B. TST)

VOLSP Es wird die Anzahl der PAM-Seiten ausgegeben, die die jeweilige Datei auf einer vorgegebenen Platte (VSN) belegt hat.

Selektionsparameter (obligatorisch): *vsn* [>*x* | <*x* | =*x*]
Mit dem optionalen Zusatz >|<|=*x* werden alle Dateien ausgewählt, die auf dem angegebenen Datenträger mehr als, weniger als oder genau *x* Seiten belegt haben.

WORK Die User Option zeigt, ob die Datei in einem SM-Pubset den Status einer Arbeitsdatei hat und daher von der Systembetreuung zu einem von ihr festgelegten Zeitpunkt gelöscht werden darf. Die User Option steht ab ab OSD V3.0 zur Verfügung.

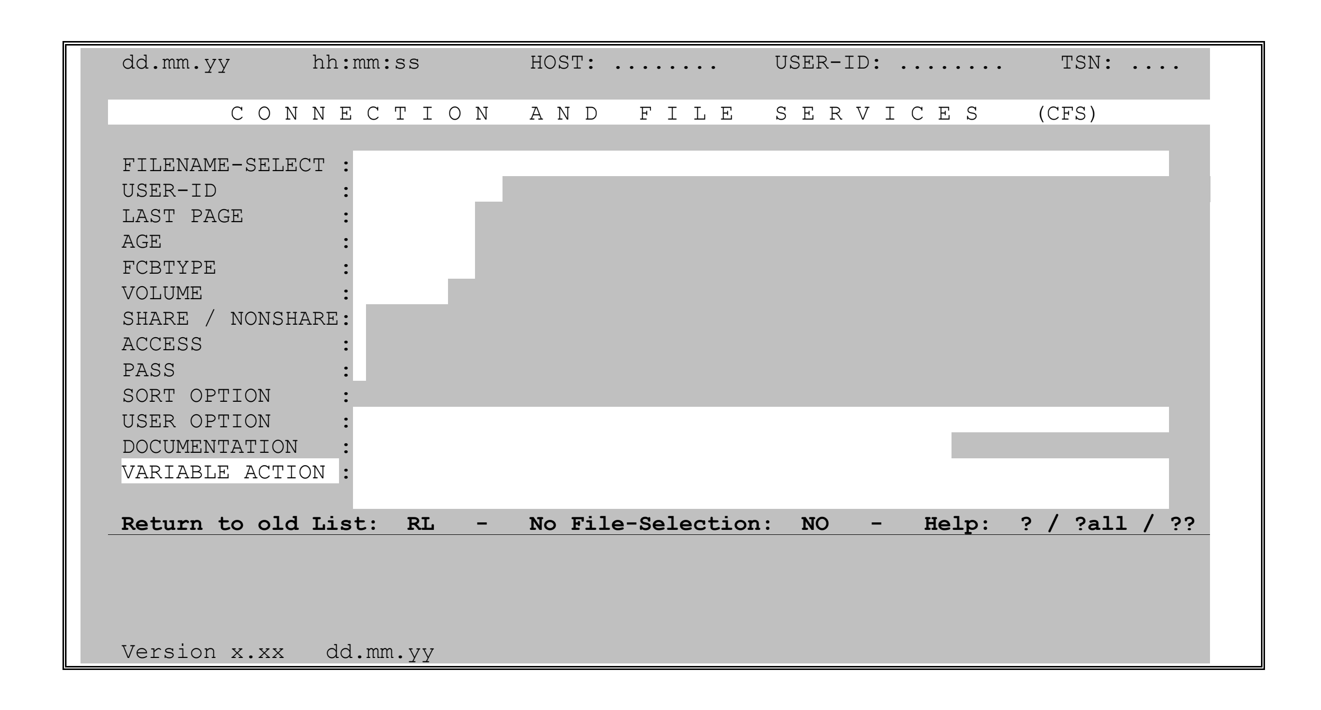
Mit der Variablen Action ONXSET WORK (siehe Seite **Fehler! Textmarke nicht definiert.**) bzw. mit dem Parameter WORK-FILE im Kommando /MODIFY-FILE-ATTR kann der Workfile-Status einer Datei festgelegt werden.

Selektionsparameter: Y | N

WRCNT Write-Count für Dateien, Jobvariable und PLAM-Bibliothekselemente: Gibt an, wie oft das Datenobjekt seit dem Erstellungszeitpunkt modifiziert wurde.
Bei Dateien ist der Write-Count als ein Byte langer Zähler realisiert, der nach dem Wert 255 wieder bei 1 beginnt.

Selektionsparameter: >*x* | <*x* | =*x*

5. Variable Actions



```
dd.mm.yy      hh:mm:ss        HOST: ........    USER-ID: ........    TSN: ....

        C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -  No File-Selection:  NO   -   Help:  ? / ?all / ??

Version x.xx   dd.mm.yy
```

Als Variable Actions bezeichnen wir Verarbeitungsoptionen für Datenobjekte (Dateien/Jobvariablen/ Bibliothekselemente), für die eine mehr oder weniger umfangreiche Parametrisierung notwendig ist. Variable Actions sind Verarbeitungen, zu deren Steuerung neben dem Namen des Datenobjekts noch eine Reihe weiterer Angaben (Parameter) erforderlich sind. Ein einfaches Beispiel ist das Übertragen von Dateien von einer Magnetplatte auf eine andere. Hierbei ist neben dem Dateinamen noch die Angabe der Volume-Serial-Nummer (VSN) sowie des Device-Typs der Magnetplatte notwendig.
Die Variablen Actions sind besonders nützlich, wenn für eine größere Menge von Datenobjekten eine komplexe Verarbeitungsoption auszuführen ist. Für Batch-Anwendungen von CFS eignen sich die in der Form ON& ... definierten Variablen Actions.

Das Gegenstück zu den Variablen Actions sind die mit einem festen Action-Code versehenen Verarbeitungsoptionen, die durch Angabe eines 1- bis 5-stelligen sog. Action-Codes bei dem Namen der gewünschten Datei aktiviert werden. Z.B.: P (= Print), PSE (= Print ,Space=E,Erase), E (= Erase), ED (= Erase, Destroy), usw.

Für diese Verarbeitungsoptionen ist außer dem Namen des zu bearbeitenden Datenobjekts keine weitere Parametrisierung notwendig. Alle verfügbaren Parameter können hier als Buchstabenzusätze an den Action-Code angehängt werden.

Bei Aktionen, die nur auf wenige Datenobjekte angewendet werden, sind die Action-Codes einfacher zu handhaben. Zudem werden Action-Codes normalerweise sofort nach Absenden der Bildschirm-Maske ausgeführt. Variable Actions werden zuerst gesammelt und zu einem späteren Zeitpunkt ausgeführt. Außerdem fällt der Schritt des Definierens der Variablen Action weg.

Alle Variablen Actions beginnen in der Form: **ONX...** oder **ON&...**

ONX... Die Variable Action wird nur auf diejenigen Datenobjekte angewendet, die der Benutzer explizit mit dem Action-Code X bzw. XE/X*t* (siehe Seite 205) angekreuzt hat.

ON&... Die Variable Action wird automatisch auf alle in der Dateienliste aufgeführten Datenobjekte angewendet, ohne daß der Action-Code X angegeben werden muß.

Aus Gründen der übersichtlichen Darstellung werden die Variablen Actions nur in der Form ONX... beschrieben.

Einige der im folgenden aufgeführten Variablen Actions stehen in der zweiten CFS-Maske (Dateienliste) auch als Einzelkommandos bzw. in Form eigener Action-Codes zur Verfügung.

Die Variablen Actions können auch im Kommandofeld der zweiten CFS-Maske eingegeben werden.

Variable Actions und Parameter Sets

Alle Variablen Actions werden in der Form ONX *varact* [*params*] oder ON& *varact* [*params*] definiert. *varact* ist hierbei der Name der variablen Action. *params* sind optionale Parameter der variablen Action.

Beispiele:

```
ONX/MOD-FILE-ATTR PROTECT=*PAR(ACCESS=*READ)
ON&FT HOST21,TSOS,ADMINSTR,C'....',DO.SUCCESS,*,DO.FAIL,*
```

Anstelle von ONX *varact* [*params*] kann die Variable Action auch in der Form ONX %*name* [*,jrsave*] angegeben werden. %*name* ist hierbei der Name eines Parameter Sets, unter dem die Variable Action zuvor gesichert wurde. Siehe hierzu Kommando SP VAR%*name* [*,jrsave*] in der Dateienliste von CFS. *jrsave* ist der Name einer von der Standard JRSAVE-Datei abweichenden Datei.
Durch ONX %*name* [*,jrsave*] ,? wird der Inhalt der in dem Parameter Set gespeicherten Variablen Action angezeigt.
Durch ONX %? werden alle Parameter Sets angezeigt, in denen Variable Actions gespeichert sind.

Die Definition einer Variablen Action über einen Parameter Set ist besonders nützlich bei umfangreichen SDF-Kommandos, die mit Hilfe des geführten SDF-Dialogs erstellt wurden.

Beispiel:

```
ON&/MOD-FILE-ATTR?
```
geführter SDF-Dialog. Es können die Parameterstrukturen des Kommandos ergänzt werden.

`SP VAR%MFA,CFS.JRSAVE.TEST`

Das im geführten SDF-Dialog ergänzte SDF-Kommando mit Parametern wird mit Hilfe des CFS-Kommandos in einem Parameterset mit Namen %MFA in der Datei CFS.JRSAVE.TEST gespeichert.

Bei einem späteren CFS-Lauf kann das vollständige SDF-Kommando wieder aktiviert werden durch Eingabe der Variablen Action: `ONX%MFA,CFS.JRSAVE.TEST`
Mit `ONX%MFA,CFS.JRSAVE.TEST,?` wird ein SDF-Fragebogen aufgeklappt, in dem das in dem Parameter Set gesicherte Kommando zusammen mit den eingegebenen Parametern angezeigt wird.

Namenstransformationsregeln '*str1*'='*str2*' | '*str1*'=DS[*n*][I] | '*str1*'=DP[*n*][I]

Alle Variablen Actions der Form ONX...'*str1*'='*str2*', die aus *name-1* eines Datenobjekts *name-2* bilden, gehen nach einer der folgenden Regeln vor:

'*str1*'='*str2*' In *name-1* wird das erste Vorkommen der Zeichenfolge '*str1*' gesucht und durch '*str2*' ersetzt. Das Ergebnis ist *name-2*.

'*str1*'=DP[*n*][I] Delete Prefix: In *name-1* wird das erste Vorkommen der Zeichenfolge '*str1*' gesucht. Alle (=DP) bzw. die nächsten *n* links von '*str1*' stehenden Zeichen (=DP*n*) werden gelöscht. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht. Das Ergebnis wird in *name-2* übertragen.

'*str1*'=DS[*n*][I] Delete Suffix: In *name-1* wird das erste Vorkommen der Zeichenfolge '*str1*' gesucht. Alle (=DS) bzw. die nächsten *n* rechts von '*str1*' stehenden Zeichen (=DS*n*) werden gelöscht. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht. Das Ergebnis wird in *name-2* übertragen.

Enthält *name-1* die Zeichenfolge '*str1*' nicht, so wird die Variable Action für das entsprechende Datenobjekt nicht durchgeführt.

Definitionsgemäß wird der String '⊔' am Ende eines jeden Namens gefunden. Somit kann durch '⊔'='.xxx' wird einem Namen ein Suffix '.xxx' angehängt werden. Durch '⊔'=DP5 werden vom Namensende ausgehend z.B. die letzten fünf Stellen entfernt.

Der Nullstring '' wird nach Definition am Anfang eines jeden Namens gefunden. Somit kann durch ''='xxx.' einem Namen ein Prefix 'xxx.' vorangestellt werden.

Protokollierung der Dateinamen unterdrücken

Die Protokollierung der ausgeführten Variablen Actions am Bildschirm kann die Ausführungszeit beträchtlich verlängern. Diese Verzögerung macht sich vor allem bei einer großen Anzahl zu bearbeitender Dateien störend bemerkbar. Durch Setzen des Prozeßschalters 14 (/SETSW ON=14) vor der Definition einer Variablen Action der Form ON&... kann die Protokollierung ausgeschaltet werden.

Alle Datenobjekte, für die die Variable Action fehlerfrei ausgeführt wurde, werden in diesem Modus in der Dateienliste unsichtbar gemacht (Action-Code '-'). Nach Ausführung der Variablen Action für alle Dateien und nach Eingabe von U auf die Terminierungsabfrage erhält der Benutzer eine aktualisierte Dateienliste, in der nur noch die Datenobjekte aufgeführt sind, für die die Variable Action nicht erfolgreich ausgeführt werden konnte. Durch das Kommando YANK (siehe Seite 288) können die unsichtbaren Einträge der Dateienliste jedoch wieder sichtbar gemacht werden.

Massenbearbeitungen von Dateien mit einer Variablen Action können auch sehr einfach in DO- oder Enter-Prozeduren ausgeführt werden. Eine entsprechende DO-Prozedur könnte wie folgt aussehen:

```
/PROC C
/SYSFILE SYSDTA=(SYSCMD)
/SETSW ON=(1,4,11,14)
/EXEC CFS
*001 FILE=...;USER-ID=...;DOC=CFS.DOC;VAR=ON&MOVE :X:
/SETSW OFF=(1,4,11,14)
/ENDP
```

Mit der obenstehenden Prozedur werden die selektierten Dateien (FILE=...;USER-ID=...) auf den Public-Volume-Set :X: übertragen. Die Namen aller Dateien, für die die Move-Operation nicht erfolgreich durchgeführt werden konnte, werden in der Dokumentationsdatei CFS.DOC gesammelt. Durch die Eingabe OLDLIST CFS.DOC im Feld FILENAME-SELECT können diese Dateien vom Benutzer selektiert und individuell nachbearbeitet werden.

Variable Action im Dialog definieren und im Enter bzw. DO-Prozedur ausführen

CFS bietet die Möglichkeit, im Dialog definierte Variable Actions (ONX../ON&..) in einem durch das Job-Report System überwachten Enterprozeß (siehe Seite 397) automatisch ausführen zu lassen. Es besteht auch die Möglichkeit, nur eine BS2000 DO-Prozedur zu generieren, in der die Ausführung der Variablen Actions beliebig oft vom Benutzer gestartet werden kann. Die Variable Action ist im gewohnten Format in der CFS-Maske im Dialog anzugeben. Bei ONX... sind die mit der Variablen Action zu bearbeitenden Dateien mit dem Action-Code X anzukreuzen. Bei ON&... werden alle in der Dateienliste aufgeführten Dateien mit der Variablen Action bearbeitet.

Die angeforderte Variable Action wird in einem eigenen Enter-Prozeß ausgeführt, falls die Definition der Variablen Action mit dem Zusatz **,ENTER** [, AP] [/*enter-params*] abgeschlossen wird.

Für die Ausführung der Variablen Action wird eine BS2000 DO-Prozedur generiert, falls die Definition der Variablen Action mit dem Zusatz **,PROC** [=*procfile*] [, AP] abgeschlossen wird. Falls der Parameter *procfile* nicht angegeben ist, wird für die generierte DO-Prozedur ein Standardname (CFS.P.*tsn*. *hhmmss*) verwendet. Die in der DO-Prozedur angesprochene DOC-Datei wird nicht automatisch gelöscht, wie dies bei ,ENTER der Fall ist.

Der Parameter AP bewirkt, daß die Passworte für die zu bearbeitenden Dateien vor dem Start des Enter-Prozesses eingegeben werden können. Das Beenden der Passworteingabe erfolgt durch Betätigen der K1-Taste oder durch Eingabe von END [, C'*passw*']. Der wahlfreie Zusatz C'*passw*' bewirkt, daß die von CFS erzeugte Enter-Datei mit einem READ-Passwort des angegebenen Inhalts versehen wird.

Bei /*enter-params* können alle in einem Enter-Kommando gültigen Parameter angegeben werden. Beispiel: /JOB-CLASS=JCBL,TIME=100,PRIORITY=180. Für weitere Informationen zu den Enter-Parametern siehe Seite 399.

Die Ausführung der mit dem Parameter ENTER versehenen Variablen Action erfolgt in der Weise, daß zunächst eine Dokumentationsdatei mit den Namen der in der Dateienliste angekreuzten Datenobjekte und danach eine Enter-Datei mit folgendem Inhalt erstellt wird:

```
/LOGON
/SYSFILE SYSDTA=(SYSCMD)
/SYSFILE SYSOUT=CFS.SYSOUT.tsn.hhmmss
/SETSW ON=(4,11)
/EXEC CFS
*001 FILE=OLDLIST CFS.DOC.tsn.hhmmss; -
VAR=ON&....
/ER CFS.DOC.tsn.hhmmss
/STEP
/LOGOFF NOSPOOL
```

Die so erstellte Enter-Datei wird durch das Job-Report System gestartet und überwacht. Bei Beendigung des Enter-Prozesses erhält der Benutzer am Bildschirm automatisch das SYSOUT-Protokoll. Eine ausführliche Beschreibung der entsprechenden Bildschirmmaske finden Sie auf Seite 405.

Beispiele:

```
ONXREORG,ENTER
```

Die mit X angekreuzten ISAM-Dateien/Bibliotheken werden in einem Enter-Prozeß, in dem CFS aufgerufen wird, reorganisiert. Es entfallen hierdurch die bei der Ausführung des REORG notwendigerweise auftretenden Wartezeiten am Bildschirm.

```
ON&FIND' EXEC CFS','/EXEC $CFS'=W CFS.TREFFER,ENTER/TIME=500
```

Alle in der Dateienliste selektierten Dateien/Bibliotheken werden nach den angegebenen Zeichenfolgen durchsucht. Die Treffersätze werden in die Datei CFS.TREFFER geschrieben. Die Variable Action wird in einem separaten Enter-Prozeß ausgeführt, der mit dem Parameter TIME=500 gestartet wird.

```
ON&FIND' EXEC CFS','/EXEC $CFS'=W CFS.TREFFER,PROC=CFS.P.FIND
```

Alle in der Dateienliste selektierten Dateien/Bibliotheken werden nach den angegebenen Zeichenfolgen durchsucht. Die Treffersätze werden in die Datei CFS.TREFFER geschrieben. Für die Ausführung der Variablen Action wird eine DO-Prozedur mit dem Namen CFS.P.FIND erzeugt. Diese wird nicht automatisch von CFS gestartet. Der Benutzer muß die Prozedur selbst starten.

BS2000-Kommando auf Dateien/Jobvariablen anwenden

ONX/*cmd-name cmd-param*

Auf die angekreuzten Dateien/Jobvariablen wird ein BS2000-Kommando angewendet. In der Variablen Action sind lediglich der Name des Kommandos (z.B. FILE) und die Parameter des Kommandos (z.B. SPACE=-9999) anzugeben. Die beiden Teile der Variablen Action werden durch ein Blank voneinander getrennt. Zum Ausführungszeitpunkt ergänzt CFS den Datei-/JV-Namen im Kommando.

Für die Einfügung des Dateinamens durch die Variable Action wird vorausgesetzt, daß das endgültige BS2000-Kommando folgendermaßen strukturiert ist: */cmd-name datei,cmd-param*. Anders strukturierte Kommandos wie /SETJV (*jv, start,len*),*wert* können mit der Variablen Action ebenfalls bearbeitet werden. Es ist hierzu der feste String **!NAME** an der Stelle anzugeben, an der der Name des Datenobjekts eingefügt werden soll.

Hinweise:

Falls es sich bei *cmd-name* um den Namen eines SDF-Kommandos handelt, so kann durch Eingabe von ONX/*cmd-name***?** der SDF-Fragebogen für die einzelnen Parameter und Strukturen angefordert werden. Als Platzhalter für den Dateinamen wird hierbei als erster Stellungsparameter der String $.NAME an das SDF-Kommando übergeben. $.NAME darf nicht gelöscht oder geändert werden. Alle im SDF-Fragebogen komplettierten Parameter werden gespeichert und bei der Ausführung des SDF-Kommandos berücksichtigt.
Der SDF-Fragebogen kann auch zu den Parametern eines bestimmten Schlüsselworts XYZ angefordert werden: ONX/*cmd-name* ...,XYZ=***PAR(?)**

Die Parameter des ausgefüllten SDF-Fragebogens zusammen mit dem SDF-Kommandonamen können für eine spätere Wiederverwendung in einem CFS-Parameter Set %*name* gesichert werden. Dazu ist in der Dateienliste von CFS das Kommando SP VAR%*name* [*,jrsave*] anzugeben Die Variable Action mit dem SDF-Kommando samt Parameter kann dann aktiviert werden durch die Definition ONX%*name* [*,jrsave*] [,?].
Mit ? wird der in dem Parameter Set abgespeicherte Inhalt angezeigt.
Für Beispiele hierzu siehe Seite **108**.

Beispiele:

```
ONX/CAT WRPASS=NONE,RDPASS=NONE,EXPASS=NONE
```
Bei den Variablen Actions ONX/CAT... / ONX/CATJV... wird STATE=U ergänzt, wenn bei den CAT-Parametern nichts angegeben wurde.

```
ONX/MC FILE=(DEL=!NAME)
ONX/SETJV (!NAME,128,10),'28.02.1992'

ONX/ENTER TIME=1000,ERASE=YES

ONX/PRINT DEV=STATION1,ERASE

ONX/ERASE
```

Die letzten drei Kommandos können auch auf Bibliothekselemente angewendet werden:

ON&/ERASE bewirkt bei Bibliotheken einen DELETE auf alle selektierten Bibliothekselemente (interessant für CFS-Prozeduren).

ON&/PRINT ... bewirkt einen PRINT auf die selektierten Bibliothekselemente.

```
ONX/MOD-FILE-ATTR PROTECT=*PAR(ACCESS=*READ)
ONX/MOD-FILE-ATTR PROTECT=*PAR(?)
ONX/MOD-FILE-ATTR ?
```

`ONX%MFA` oder `ONX%MFA,?`

Dateien ohne Namensänderung in Bibliothek aufnehmen/Bibliothekselemente in andere Bibliothek übertragen

ONX**ADD** *bibliothek* [,CA] [,CPR] [,CL[*typ*]]

Aufnahme der mit X markierten Dateien in eine Bibliothek (FMS/LMS/PLAM).
Übertragen der mit X markierten Bibliothekselemente in eine andere Bibliothek.

bibliothek Name der Bibliothek, in der die Datenobjekte abzuspeichern sind.

Die Namen der erzeugten Bibliothekselemente ergeben sich aus den Namen der Eingabe-Dateien bzw. aus den Namen der Eingabe-Bibliothekselemente.

Bei Aufnahme in eine LMS-ISAM Bibliothek ist folgendes zu beachten: Die Elementnamen werden aus den ersten 8 Stellen der letzten Teilqualifizierung des Namens der jeweiligen Eingabedatei gebildet.

CA Copy All Delta-Versions. Dieser Parameter ist nur von Bedeutung bei der Übertragung von deltagespeicherten Elementen einer PLAM-Bibliothek in eine zweite PLAM-Bibliothek. Für eine ausführliche Beschreibung siehe Seite 116.

CPR Copy Protection. Dieser Parameter ist nur von Bedeutung bei der Übertragung von Elementen einer PLAM-Bibliothek in eine zweite PLAM-Bibliothek. Für eine ausführliche Beschreibung siehe Seite 117.

CL[*typ*] Create Library. Falls die Bibliothek noch nicht existiert, wird sie automatisch neu angelegt. Es erfolgt keine Rückfrage "Create ... Library ?". Falls die Bibliothek schon existiert, so bleibt dieser Parameter ohne Wirkung.
Mit *typ* kann der Typ der Neu anzulegenden Bibliothek bestimmt werden. Mögliche Eingaben sind: F/L/P/N. Standard: P

Beispiel für die Namensbildung der Bibliothekselemente bei Aufnahme in verschiedene Bibliotheken. Name der aufzunehmenden Datei: ABC.S.TESTDATEI

FMS	-->	ABC.S.TESTDATEI	
LMS (ISAM)	-->	TESTDATE	(8 Zeichen der letzten Teilqualifizierung)
PLAM	-->	S/ABC.S.TESTDATEI	(S = Standard Elementtyp)

Hinweise:

Mit der Variablen Action ONXADD können auch Elemente einer Bibliothek in eine andere Bibliothek übertragen werden.

Falls die angegebene Bibliothek noch nicht existiert, so fragt CFS den Benutzer, welchen Typ die neu anzulegende Bibliothek erhalten soll:

"Create FMS-/LMS-/PLAM-Library/ZIP/None ? (F/L/P<1/2>/Z/N/?)"

F Es wird eine FMS-Bibliothek eingerichtet.
L Es wird eine Bibliothek des alten LMS-Formats (OSM: Source/Macro) erzeugt.
P Es wird eine PLAM-Bibliothek eingerichtet (BLKSIZE=(STD,2)).
P1 Es wird eine PLAM-Bibliothek eingerichtet in einem mit FTP besser versendbaren Format (BLKSIZE=(STD,1)).
Z Es wird ein neues ZIP-Archiv eingerichtet.
N Der Benutzer hat sich bei der Eingabe des Bibliotheksnamen vertippt. Es soll keine Bibliothek mit diesem Namen eingerichtet werden. Die Variable Action ONXADD/das ADD-Kommando wird in diesem Fall zurückgewiesen.
? Es werden diese Hilfe Informationen angezeigt.

Falls Elemente einer Bibliothek in eine neu einzurichtende Bibliothek übertragen werden, so wird diese von CFS mit den gleichen Speicherplatz Zuweisungen wie die Originalbibliothek eingerichtet.

Es können Dateien beliebigen Formats in FMS-Bibliotheken aufgenommen werden, insbesondere auch PAM-Dateien (gebundene Programmphasen) und Dateien mit RECFORM=F,BLKSIZE=(STD,*x*). Die so aufgenommenen Elemente können mit dem Programm FMS jedoch nicht bearbeitet werden.

Falls Dateien in LMS-ISAM Bibliotheken aufgenommen werden, so sollten die Dateinamen das Zeichen '#' nicht enthalten, da CFS die folgenden Zeichen als Versionsangabe für das Bibliothekselement interpretiert.

Bei der Übertragung von Elementen einer PLAM-Bibliothek in eine FMS-Bibliothek wird den Elementnamen die Typbezeichnung von PLAM als Teilqualifizierung vorangestellt (z.B. 'S.'). Falls die Elemente ohne Typbezeichnung übernommen werden sollen, ist die Variable Action in der Form ONXADD '*tl*'='' *fms-bibl* anzugeben.

Add: Aufnehmen in Bibliothek mit Änderung des Namens

ONX**ADD** [*t*/] ['*str1*'= ...] *bibliothek* [,CA] [,CPR] [,CL[*typ*]]

Die mit X angekreuzten Dateien/Bibliothekselemente werden als Elemente in die angegebene Bibliothek aufgenommen. Bei der Übertragung zwischen zwei PLAM-Bibliotheken werden die Versionsbezeichnungen der Elemente übernommen. Bei der Übertragung aus einer LMS-ISAM Bibliothek in eine PLAM-Bibliothek werden die Versionsbezeichnungen der Elemente nicht übernommen. Bei der Aufnahme von Dateien in eine PLAM-Bibliothek werden die neuen Elemente ohne Versionsbezeichnung angelegt.

'*str1*'= ... Namenstransformationsregeln

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

'*str1*'='*str2*'
Der Name des neuen Bibliothekselements wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und durch '*str2*' ersetzt wird. Definitionsgemäß wird der String '⊔' am Ende und der Nullstring '' am Anfang eines jeden Namens gefunden. Enthält der ursprüngliche Name die Zeichenfolge '*str1*' nicht, so wird die Variable Action für dieses Datenobjekt nicht ausgeführt.

'*str1*'=DP[*n*][I]
Delete Prefix: Der Name des neuen Bibliothekselements wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der links davorstehende Namensteil bzw. die *n* links davorstehenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht.

'*str1*'=DS[*n*][I]
Delete Suffix: Der Name des neuen Bibliothekselements wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der nachfolgende Namensteil bzw. die *n* nachfolgenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht.

Fehlt die Angabe '*str1*'=..., so wird der Name des neuen Bibliothekselements 1 zu 1 aus dem Namen der Datei bzw. des ursprünglichen Bibliothekselements gebildet. Näheres zu den Namenstransformationsregeln siehe Seite **109**. Beispiele für die Namenstransformation finden Sie unten.

t/ Nur bei Aufnahme in eine PLAM-Bibliothek: Die mit X markierten Datenobjekte werden als Elemente des angegebenen Typs in die PLAM-Bibliothek aufgenommen. Fehlt die Typangabe, so wird S/ als Standardtyp ergänzt. Beim Übertragen von Elementen von PLAM nach PLAM werden bei fehlender Typangabe die Elemente mit dem gleichen Typ wie in der Ursprungsbibliothek eingerichtet.

Der Benutzer kann beim Ankreuzen der Dateien den in der Variablen Action definierten Elementtyp durch Eintragen des Action-Codes X*t* ändern. *t* bezeichnet einen von der Variablen Action abweichenden Elementtyp. Die Typangabe im Action-Code hat Vorrang vor dem in der Variablen Action festgelegten Typ. E ist als Typangabe im Action-Code nicht zulässig, da der Action-Code XE das Löschen der Ursprungsdatei bzw. des Elements nach erfolgreicher Aufnahme in die Zielbibliothek bewirkt.

bibliothek Name der Bibliothek, in der die Datenobjekte abzuspeichern sind.

CA Copy All Delta-Versions. Dieser Parameter ist nur von Bedeutung bei der Übertragung von deltagespeicherten Elementen einer PLAM-Bibliothek in eine zweite PLAM-Bibliothek. Deltagespeicherte Elemente werden strukturerhaltend, d.h. mit allen

zugehörigen Versionen in die neue Bibliothek übertragen. Ohne Angabe der CA-Option (Standardfall) wird nur die in der Dateienliste mit X markierte Elementversion in die neue Bibliothek übertragen.

CPR — Copy Protection Attributes. Dieser Parameter ist nur von Bedeutung bei der Übertragung von Elementen einer PLAM-Bibliothek in eine zweite PLAM-Bibliothek unter der eigenen Benutzerkennung. Evtl. vorhandene Read-, Write- und Execute-Passworte sowie die Basic Access Control Listen (BACL) der Ursprungselemente werden in die kopierten Elemente übernommen. Falls die Bibliothek-2 neu eingerichtet wird, werden auch die Bibliotheks- und Typattribute aus der Ursprungsbibliothek übernommen. Zu den Bibliotheksattributen gehören die Administrationsrechte (siehe Seite **197**) und die Standard-Schutzattribute für neue Elemente (siehe Seite **204**). Zu den Typattributen gehören die Standard-Schutzattribute für neu hinzukommende Elemente dieses Typs.

CL[*typ*] — Create Library. Falls die Bibliothek noch nicht existiert, wird sie automatisch neu angelegt. Es erfolgt keine Rückfrage "Create ... Library ?". Falls die Bibliothek schon existiert, so bleibt dieser Parameter ohne Wirkung.
Mit *typ* kann der Typ der Neu anzulegenden Bibliothek bestimmt werden. Mögliche Eingaben sind: F/L/P[1]/N. Standard: P

Hinweise:

Durch die Typangabe P1 wird eine PLAM-Bibliothek mit BLKSIZE=(STD,1) erzeugt. Standard ist (STD,2).

Bei der Aufnahme von Dateien in eine PLAM-Bibliothek bzw. bei der Übertragung von Bibliothekselementen in eine andere PLAM-Bibliothek erbt das neue Element das Alter des Ursprungsobjekts.

Beim Übertragen von Elementen einer PLAM-Bibliothek in eine andere PLAM-Bibliothek sind folgende Sonderfälle zu beachten:
ONXADD '⊔'='*suffix*' *bibliothek* — ein Blank in String-1
Bei Elementen mit Versionen wird die angegebene Suffix nach dem Elementnamen und vor der Versionsbezeichnung eingefügt.
ONXADD '⊔'='*/suffix*' *bibliothek* — ein Blank in String-1
Bei Elementen mit Versionsbezeichnung wird diese entfernt und durch die angegebene Version /suffix ersetzt.
Bei Elementen ohne Version wird die angegebene Version /suffix an den Elementnamen angehängt.
ONXADD '⊔ ⊔'='*suffix*' *bibliothek* — zwei Blanks in String-1
Bei Elementen mit Versionen wird die angegebene Suffix an die bestehende Versionsbezeichnung angehängt.

CFS kann bei Aufnahme von Elementen in eine PLAM-Bibliothek den angegebenen Elementtyp auf LMS-Kompatibilität prüfen. Es stehen die folgenden zwei Möglichkeiten zur Auswahl:

a) nicht LMS-kompatible Typbezeichnungen werden abgewiesen

b) der Benutzer wird gefragt, ob er wirklich ein Element mit einer nicht LMS-kompatiblen Typbezeichnung in die Bibliothek aufnehmen möchte.

Zur Steuerung der Typ-Überprüfung stehen folgende Kommandos zur Verfügung:

CT	Check Types. Nur LMS-kompatible Typangaben zugelassen.
CTQ	Check Types Query. Bei nicht LMS-kompatiblen Typangaben wird der Benutzer gefragt (Standard).
NCT	No Check Types. Keine Typ-Überprüfung.

Mit LMS verträgliche Elementtypen in PLAM-Bibliotheken:

C Coreimage (mit TSOSLNK gebundene Programme)
D Document/Data
H von H-Assembler erzeugte Daten
J JCL (DO-/Enter-Prozeduren)
L von Binder des BS2000 erzeugte Daten
M Macro
P Print (mit SPACE=E ausdruckbare Listdateien)
R Relocatable (Objektmodule)
S Source
X Extra

Beispiele für die Namensbildung der Bibliothekselemente bei Aufnahme einer Datei in verschiedene Arten von Bibliotheken. Dateiname = ABC.S.TESTDATEI.

`ONXADD'S.'='SRC.' bibl`

FMS	ABC.S.TESTDATEI -->	ABC.SRC.TESTDATEI
LMS (ISAM)	ABC.S.TESTDATEI -->	TESTDATE (8 Stellen der letzten Teilqual.)
PLAM	ABC.S.TESTDATEI -->	S/ABC.SRC.TEST

`ONXADD'ABC.S.'='' bibl`

FMS	ABC.S.TESTDATEI -->	TESTDATEI
LMS (ISAM)	ABC.S.TESTDATEI -->	TESTDATE
PLAM	ABC.S.TESTDATEI -->	S/TESTDATEI

`ONXADD'TEST'=DS,bibl`

FMS	ABC.S.TESTDATEI -->	TESTDATEI
LMS (ISAM)	ABC.S.TESTDATEI -->	TESTDATE
PLAM	ABC.S.TESTDATEI -->	S/TESTDATEI

`ONXADD''='XYZ' bibl`

FMS	ABC.S.TESTDATEI -->	XYZABC.S.TESTDATEI
LMS (ISAM)	ABC.S.TESTDATEI -->	TESTDATE
PLAM	ABC.S.TESTDATEI -->	S/XYZABC.S.TESTDATEI

`ONXADD' '='.301192' bibl`

FMS	ABC.S.TESTDATEI -->	ABC.S.TESTDATEI.301192
LMS (ISAM)	ABC.S.TESTDATEI -->	301192
PLAM	ABC.S.TESTDATEI -->	S/ABC.S.TESTDATEI.301192

`ONXADD' '=DP5,bibl`

FMS	ABC.S.TESTDATEI -->	ABC.S.TEST
LMS (ISAM)	ABC.S.TESTDATEI -->	TEST
PLAM	ABC.S.TESTDATEI -->	S/ABC.S.TEST

`ONXADD bibl` oder `ONXADD''='' bibl`

FMS	ABC.S.TESTDATEI -->	ABC.S.TESTDATEI
LMS (ISAM)	ABC.S.TESTDATEI -->	TESTDATE
PLAM	ABC.S.TESTDATEI -->	S/ABC.S.TESTDATEI

```
ONXADD X/bibl
```
FMS ABC.S.TESTDATEI --> ABC.S.TESTDATEI
LMS (ISAM) ABC.S.TESTDATEI --> TESTDATE
PLAM ABC.S.TESTDATEI --> **X**/ABC.S.TESTDATEI

```
ONXADD X/'ABC.S.'='' bibl
```
FMS ABC.S.TESTDATEI --> TESTDATEI
LMS (ISAM) ABC.S.TESTDATEI --> TESTDATE
PLAM ABC.S.TESTDATEI --> **X**/TESTDATEI

Änderung des Elementtyps bei Übertragung zwischen zwei PLAM-Bibliotheken.

```
ONXADD M/bibl
```
PLAM **S**/TEST --> **M**/TEST

```
ONXADD'S/'='M/' bibl
```
PLAM **S**/TEST --> **M**/TEST

`ONXADD bibl` und Ankreuzen des zu übertragenden Bibliothekselements mit dem Action-Code X*t*, wobei *t* der Typ des Elements in der Zielbibliothek ist.
PLAM **S**/TEST --> **M**/TEST
XM = Action-Code

Ändern der Versionsbezeichnung beim Übertragen von PLAM-Elementen:

```
ONXADD'/###'='' bibl
```
Die Anzahl der anzugebenden Zeichen '#' richtet sich nach der Stellenanzahl in der Versionsbezeichnung.
PLAM S/TEST/017 --> S/TEST

```
ONXADD'/###'='/001' bibl
```
PLAM S/TEST/017 --> S/TEST/001

Voranstellen eines Prefix im Elementnamen:

```
ONXADD'/'='/ABC.' bibl
```
PLAM S/TEST --> S/ABC.TEST

Übertragen von LMS-ISAM Elementen nach PLAM mit gleichzeitiger Vorgabe einer Versionsbezeichnung:

```
ONXADD M/' '='/001' bibl
```
LMS-ISAM MAC1 /../... --> M/MAC1/001 (PLAM)

Delta-gespeicherte PLAM-Elemente:

Delta-Speicherung bedeutet, daß verschiedene Versionen des gleichen Elements in einer Form abgelegt sind, bei der nur die geänderten Sätze bezüglich der Vorgängerversion gespeichert werden. Durch die Delta-Technik kann eine bedeutende Reduzierung des benötigten Speicherplatzes erreicht werden.

Wird ein Element aus einer anderen Bibliothek oder eine Datei mit ONXADD in eine PLAM-Bibliothek übertragen, so sind bei der Delta-Speicherung verschiedene Fälle zu unterscheiden:

1) In der Variablen Action ONXADD wurde der Parameter CA (Copy All) angegeben. In diesem Fall werden bei einem deltagespeicherten Element auto-

matisch alle Deltaversionen strukturerhaltend in die Zielbibliothek übertragen. Die folgenden Punkte 2) und 3) beziehen sich auf den Fall, daß die CA-Option nicht angegeben wurde (Standard).

2) In der Zielbibliothek existiert bereits ein Element des angegebenen Namens und Elementtyps. In diesem Fall wird die Speicherungsform des bereits bestehenden Elements beibehalten. Dies bedeutet:

a) Es liegt ein Element als Voll-Element (in nicht delta-gespeicherter Form) vor. Das bisherige Element wird durch das neu aufzunehmende Element gleichen Namens überschrieben, sofern nicht verschiedene Versionsnummern vorliegen.

b) Es liegt ein Element in delta-gespeicherter Form vor. Das neu aufzunehmende Element (die neu aufzunehmende Datei) wird als eine neue Version durch Ergänzung um die nächsthöhere Versionsnummer an den bestehenden Delta-Baum angehängt. Das alte Element wird dadurch nicht überschrieben und steht weiterhin zur Verfügung.

3) In der Zielbibliothek existiert kein Element des angegebenen Namens. In diesem Fall sind folgende Möglichkeiten gegeben:

a) Das zu übertragende Element ist nicht delta-gespeichert. Das neue Element wird in der Zielbibliothek als Vollelement angelegt.

b) Das zu übertragende Element liegt in der Ursprungsbibliothek in delta-gespeicherter Form vor.

- Das Element wird in der Zielbibliothek in nicht delta-gespeicherter Form angelegt, falls die User Option DELTA bei der Selektion der Ursprungsbibliothek nicht angegeben wurde.

- Das Element wird in der Zielbibliothek in delta-gespeicherter Form angelegt, falls die User Option DELTA bei der Selektion der Elemente der Ursprungsbibliothek angegeben wurde. Die ausführliche Beschreibung der User Option DELTA finden Sie auf Seite 82.

Dateien / Jobvariablen mit ARCHIVE sichern

ONX**ARCHIVE** *anw* [PARAM *param*]

Es wird eine DO-Prozedur für einen ARCHIVE-Lauf erstellt.

anw ARCHIVE-Verarbeitungsanweisung. Z.B. E[XPORT], S[AVE].

param zusätzliche Parameter für den ARCHIVE-Lauf. Bei Angabe von Parametern werden die von CFS standardmäßig verwendeten Archive-Parameter PARAM CNS=YES|NO, UNLOAD=YES|NO,STREAM=YES|NO außer Kraft gesetzt und müssen bei Bedarf explizit angegeben werden.

Nach Erstellung der DO-Prozedur für den ARCHIVE-Lauf, wird der Benutzer gefragt, ob diese Prozedur im Dialog, im Enter oder garnicht angestartet werden soll (nur Generierung einer DO-Prozedur).

Bei Ablauf der DO-Prozedur im Dialog wird in CFS ein DO-Kommando abgesetzt und das Programm wird damit beendet.

Der Ablauf der DO-Prozedur im Enter wird über die Job-Report Funktion realisiert, d.h. bei Beendigung des Enter-Auftrags wird der Benutzer automatisch benachrichtigt und erhält das SYSOUT-Listing am Bildschirm.

Beispiele:

```
ONXARCHIVE E L=SYSOUT,TAPES=888888
```

```
ONXARCHIVE S NOW=NO,CH=NO,ER=YES,TAPES=123456,DEVICE=TAPE-
C4 PARAM CAT-ID=YES,UNLOAD=YES,STREAM=YES
```

Hinweis:

ARCHIVE kann mit der Anweisung PARAM CAT-ID=YES|NO so parametrisiert werden, daß die Dateinamen mit bzw. ohne CAT-IDs gesichert werden.
Falls bei der Variablen Action ONXARCHIVE keine Anweisung PARAM CATID=... vom Benutzer angegeben wurde, so gilt folgendes:
Bei eingeschalteter CFN-Option (Complete Filename) oder falls aus einem anderen Grund die Dateinamen in der Dateienliste mit CAT-IDs angezeigt werden (:x:$user-id. datei-name), so fügt CFS in die Archive-Prozedur die Anweisung PARAM CATID=YES ein. Die Angabe der User Option NOC (NO Cat-ID) bewirkt, daß die Cat-IDs aus den Dateinamen entfernt werden. Beispiel: NP ...;U=NOC

Dateien / Bibliothekselemente in anderes Datenformat konvertieren

ONX**CONV** ['*str1*'= ... | PRE=*prefix* | SUF=*suffix*] ,*f-attr* | ? [, *opt*]

Die mit dem Action-Code X gekennzeichneten Dateien/Bibliothekselemente werden in ein anderes Datenformat konvertiert. Die Namen der konvertierten Datenobjekte werden gebildet aus den Namen der Eingabe-Dateien/Bibliothekselemente nach einer der nachfolgend aufgeführten Regeln.

'*str1*'= ... Namenstransformationsregeln

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

'*str1*'='*str2*'
Die Zeichenfolge '*str1*' wird durch '*str2*' ersetzt.

'*str1*'=DP[*n*][I]
Vor der Zeichenfolge '*str1*' werden alle bzw. *n* Stellen gelöscht.

'*str1*'=DS[*n*][I]
Nach der Zeichenfolge '*str1*' werden alle bzw. *n* Stellen gelöscht (näheres siehe Seite **109**).

PRE=*prefix*
Ein Prefix wird vorangestellt, z.B. PRE=CONV.

SUF=*suffix*
Ein Suffix wird dem Namen angehängt, z.B. SUF=.CONV

Falls in der CONV-Anweisung weder '*str1*'=..., noch PRE=*prefix*, noch SUF=*suffix* angegeben wurde, so wird die Eingabedatei/das Eingabeelement in eine Hilfsdatei umgesetzt und anschließend auf das ursprüngliche Datenobjekt zurückkopiert. Die Hilfsdatei wird danach gelöscht.

f-attr Attribute der Ausgabedatei. Hier wird das Datenformat der Ausgabedatei gemäß der Syntax eines FILE- oder ADD-FILE-LINK Kommandos festgelegt.

Beispiel (Parameter gemäß FILE-Kommando):
```
FCBTYPE=ISAM,KEYLEN=20,KEYPOS=8,BLKSIZE=(STD,16),
SPACE=(64,64)
```

Beispiel (Parameter gemäß ADD-FILE-LINK Kommando):
```
ACCESS=*ISAM,SUPP=*DISK(ISAM=(KEY-LEN=20,KEY-POS=8)),BUFFER-
L=*STD(16)
```

? Es wird der SDF-Fragebogen des Kommandos /ADD-FILE-LINK (ADFL) ausgegeben. Die vorgegebenen Werte der ersten beiden Parameter (LINK-NAME=OUTFCB und FILE-NAME=CFS...) dürfen nicht verändert oder gelöscht werden.

Zur Umsetzung der Datenstruktur der Dateien ist folgendes anzugeben:

BLKCTRL= PAMKEY | DATA | DATA4K | NO
Beispiel: `ONXCONV BLKCTRL=DATA`
Bei Programmphasen und PLAM-Bibliotheken wird die Variable Aktion ONXCONV BLKCTRL=NO an das Dienstprogramm PAMCONV zur Ausführung weiter gereicht.

opt eine der folgenden Angaben oder eine Kombination:

IK	Insert Keys
EK	Eliminate Keys
NA	Non Ascending Keys
S	Same attributes
LEN=*n*	logische Satzlänge
Füllmuster	C'*string*' | X'*string*'

Füllmuster Vorgabe einer Zeichenfolge, mit der die Ausgabesätze aufgefüllt werden, falls diese eine größere Länge haben als die Sätze der Eingabedatei. Standard: C'⊔'.

IK Insert Keys: Für die Sätze der Ausgabedatei werden von CFS selbständig ISAM-Schlüssel generiert. Diese Schlüsselfelder können mit der Option IK auch bei Ausgabedateien vom Fcbtyp SAM erzeugt werden. In diesem Fall erhalten die Ausgabesätze nach dem Satzlängenfeld eine 8-stellige Nummerierung.

Standard: kein IK, d.h. der ISAM-Schlüssel ist in den Daten der zu konvertierenden Datei enthalten. Falls die ISAM-Schlüssel nicht in aufsteigender Reihenfolge vorliegen, so ist die Option NA (Non Ascending Keys, siehe unten) anzugeben.

EK Eliminate Keys: Bei der Konvertierung von ISAM nach SAM wird der ISAM-Schlüssel aus dem Eingabesatz ausgeblendet.

Standard: kein EK, d.h. bei Eingabedateien vom Typ ISAM werden die Keys beim Konvertieren nicht ausgeblendet.

NA Non Ascending Keys: Bei Konvertierung in eine ISAM-Datei enthalten die Eingabesätze die Daten für die ISAM-Schlüssel. Diese liegen jedoch nicht in aufsteigender Reihenfolge vor, so daß CFS mit dem ISAM-Makro STORE anstelle von PUT arbeiten muß. Die Angabe NA bewirkt eine Umsortierung der Sätze der Eingabedatei.

Standard: kein NA. Eine nicht aufsteigende Reihenfolge des Schlüsselbegriffs in der Eingabedatei führt auf einen DVS-Fehler.

S — Same attributes: Die Ausgabedateien erhalten die in ONXCONV... angegebenen Dateiattribute. Alle nicht angegebenen Dateiattribute werden von den jeweiligen Eingabedateien übernommen.
Die Option S muß stets als letzte Option der Variablen Action CONV angegeben werden. Die S-Option kann ab BS2000-Version 9.5 verwendet werden.

Standard: Für alle bei ONXCONV ... nicht angegebenen Dateiattribute werden die Standards des BS2000 angenommen: FCBTYPE= ISAM, RECFORM=V, BLKSIZE=(STD,1), KEYPOS=5, KEYLEN=8.

LEN=*n* — gibt die Gesamtlänge der Ausgabesätze an. Um die angegebene Ausgabelänge zu erreichen, werden die Sätze rechts mit Blanks aufgefüllt. Die vier Byte für das Satzlängenfeld werden bei der Angabe LEN= mitgezählt. Die RECSIZE im Katalog wird dadurch nicht beeinflußt und hat bei Dateien variabler Satzlänge den Wert RECSIZE=0.

Standard: Bei Ausgabedateien mit RECFORM=V ergibt sich die Satzlänge aus der Länge der Eingabesätze.

Hinweise:

Falls die Ausgabedatei im Format RECFORM=U gewünscht wird, muß ebenfalls der Parameter RECSIZE=8 angegeben werden (Mehrzweckregister 8 wird im Programm für die Länge der Ausgabesätze verwendet).

Bei der Angabe des Dateieiformats der Ausgabedateien ist zu beachten, daß für jedes nicht angegebene Dateiattribut der entsprechende Standardwert des File-Kommandos eingesetzt wird. Falls keine Dateieigenschaften angegeben wurden, so werden folgende Standardwerte angenommen:
FCBTYPE=ISAM,KEYPOS=5,KEYLEN=8,RECFORM=V,BLKSIZE=(STD,1).
Zum gleichen Thema siehe auch Beispiel 2 weiter unten.

Eine Ausnahme bei den Standardwerten der Dateiattribute bildet der Parameter BLKCTRL. Falls die angegebenen Dateiattribute mit BLKCTRL= beginnen, so erhalten die umgesetzten Dateien bis auf die Werte für BLKCTRL und PAD (falls angegeben), exakt die gleichen Dateieigenschaften wie die Eingabedateien. Siehe hierzu auch Beispiel 3 weiter unten.

Mit der Variablen Action ONXCONV können auch SAM-/ISAM-Dateien in PAM-Dateien umgesetzt werden und umgekehrt. Es sind dabei folgende Fälle zu unterscheiden:

SAM/ISAM --> PAM (ONXCONV .... , FCBTYPE=PAM)

1) Eingabedatei: BLKSIZE = (STD,1)

   Die Eingabedatei (SAM-/ISAM-Datei) wird PAM-Blockweise gelesen und als PAM-Datei in die Ausgabedatei geschrieben. Es werden dabei keinerlei Dateninhalte verändert. In diesem Fall wird lediglich der FCBTYPE im Katalog in PAM umgewandelt.

2) Eingabedatei: BLKSIZE > (STD,1)

   Es wird unterstellt, daß die ersten 16 Bytes der Eingabesätze der SAM-/ ISAM-Datei (Recform=V) die Daten des PAM-Keys enthalten. Danach werden 2048 Byte an Daten für jeweils einen PAM-Block erwartet.

PAM --> SAM/ISAM (ONXCONV .... , FCBTYPE=SAM/ISAM)

1) Ausgabedatei: BLKSIZE = (STD,1)
ONXCONV ... ,FCBTYPE=SAM/ISAM,BLKSIZE=(STD,1)

Die Eingabedatei wird PAM-Blockweise in die Ausgabedatei übertragen. Es werden dabei keinerlei Dateninhalte verändert. In diesem Fall wird lediglich der FCBTYP im Katalog von PAM nach SAM/ISAM umgewandelt.

2) Ausgabedatei: BLKSIZE > (STD,1)
z.B.: ONXCONV ... ,FCBTYPE=SAM/ISAM,BLKSIZE=(STD,2)

In den ersten 16 Byte jedes Ausgabesatzes wird der PAM-Key des Eingabe-Blocks abgelegt. Anschließend werden die 2048 Byte Daten des gelesenen PAM-Blocks abgelegt.

Beispiele:

`ONXCONV SUF=.F80,RECFORM=F,RECSIZE=80,FCBTYPE=SAM`
Die mit dem Action-Code X angekreuzten Dateien werden in SAM-Dateien mit RECFORM=F,RECSIZE=80 konvertiert. Die Namen der umgesetzten Dateien werden gebildet aus den Namen der Eingabedateien, wobei das Suffix '.F80' an den Namen angehängt wird.

`ONXCONV PRE=#,IK,LEN=300`
Die mit dem Action-Code X angekreuzten Dateien (Fcbtyp SAM oder ISAM) werden in ISAM-Dateien mit dem Standardformat des File-Kommandos: FCBTYPE=ISAM,RECFORM=V,KEYPOS=5,KEYLEN=8 konvertiert. Die ISAM-Schlüssel werden (zusätzlich zu evtl. schon vorhandenen Schlüsseln) neu generiert (IK). Die Sätze der Ausgabedateien werden mit Blanks auf 300 Byte aufgefüllt. Sind Datensätze der Eingabedatei länger, so werden diese rechts abgeschnitten. Die Namen der Ausgabedateien werden aus den Namen der Eingabedateien und Voranstellung des Zeichens # gebildet (temporäre Datei).

`ONXCONV BLKCTRL=DATA`
Die mit dem Action-Code X angekreuzten Dateien werden in das interne Format BLKCTRL=DATA übergeführt. Die Eingabedateien werden nach erfolgreicher Ausführung der Operation durch die konvertierten Dateien überschrieben. Im Unterschied zum letzten Beispiel werden alle Attribute der Eingabedateien außer BLKCTRL automatisch, d.h. auch ohne Angabe der S-Option in die Ausgabedateien übernommen.

`ONXCONV' '='.SAM',FCBTYPE=SAM,LEN=80`
Die mit dem Action-Code X angekreuzten Dateien werden in Dateien mit FCBTYPE =SAM,RECFORM=V,BLKSIZE=(STD,1) konvertiert. Die Namen der Ausgabedateien werden gebildet, indem das Suffix '.SAM' an die Namen der Eingabedateien angehängt wird.

`ONXCONV' '='.SAM',FCBTYPE=SAM,EK,S`
Die mit dem Action-Code X angekreuzten Dateien werden in SAM-Dateien konvertiert. Im Unterschied zum letzten Beispiel erhalten die Ausgabedateien lediglich die Eigenschaft FCBTYPE=SAM. Die Werte für RECFORM, RECSIZE und BLKSIZE werden aus den jeweiligen Eingabedateien übernommen (S: Same attributes). Falls die Eingabedateien vom Fcbtyp ISAM sind, so werden die ISAM-Schlüssel bei der Konvertierung in SAM-Dateien aus den Datensätzen entfernt (EK: Eliminate Keys).

`ONXCONV'.D.'='.D.NEU.',FCBTYPE=ISAM,KEYPOS=200, KEYLEN=20,IK`
Die mit dem Action-Code X angekreuzten Dateien werden in ISAM-Dateien konvertiert. Die Namen der Ausgabedateien werden gebildet, indem in den Namen der Eingabedateien die Zeichenfolge '.D.' gesucht und durch '.D.NEU.' ersetzt wird. Findet sich in dem Namen einer durch X angekreuzten Datei die Zeichenfolge '.D.' nicht, so wird diese Datei nicht konvertiert.

Die erzeugten ISAM-Dateien werden mit KEYPOS=200 und KEYLEN=20 angelegt. Die Daten für die ISAM-Schlüssel werden von CFS selbst generiert und müssen in den sequentiellen Eingabedateien nicht vorhanden sein (IK: Insert Keys). Bei der Generierung der Schlüssel für die Ausgabedateien verfährt CFS nach folgender Regel: Die ersten 8 Stellen des Schlüssels sind numerisch (Schrittweite=10). Falls die zu erzeugenden Schlüssel eine größere Länge als 8 Byte haben, so werden die Stellen danach auf binär Null gesetzt.

Aufteilen einer sequentiellen Datei in kleinere, gleich große Teildateien

1) Ausgangsdatei mit dem Action-Code D (Display) markieren.

2) Kommando PAM wählen (Datei im PAM-Format darstellen). Damit wird es möglich, physikalische PAM-Blöcke der Eingabedatei in eine andere Datei zu übertragen. Im normalen Anzeigemodus können nur einzelne Datensätze in eine andere Datei geschrieben werden.

3) Anwendung des Write-Kommandos: W*n*, *subdatei*. Es werden fortlaufend *n* PAM-Seiten der Eingabedatei in verschiedene Subdateien mit FCBTYPE=PAM geschrieben. Die Subdateien müssen unterschiedliche Namen erhalten.

4) Rückkehr in die Selektionsmaske von CFS (Kommando NP) und Auswahl aller erzeugten Subdateien.

5) Rückkonvertierung der Subdateien von FCBTYPE=PAM --> SAM:
ONXCONV SUF=.SAM,FCBTYPE=SAM

PLAM: Bibliothekselemente in delta-gespeicherte Form konvertieren

ONX**CONV** DELTA=*vers*

Die durch den Action-Code X gekennzeichneten PLAM-Bibliothekselemente werden in Basiselemente für Delta-Speicherung angelegt.

vers 1- bis zehnstellige numerische Versionsbezeichnung des Delta-Elements.
Bei jeder nachfolgenden Modifikation des Bibliothekselements wird die Versionsbezeichnung von CFS automatisch um 1 erhöht. Es ist daher sicherzustellen, daß für die anfängliche Version genügend viele Stellen reserviert werden. Z.B. ONXCONV DELTA=0001. Das so eingerichtete Delta-Element kann 9999 mal geändert werden.

Hinweise:

Es können nur solche PLAM-Elemente in Delta-Form konvertiert werden, die keine Format-B Sätze enthalten. Nicht konvertiert werden können z.B. gebundene Programmphasen (TYP=C) und als Bibliothekselemente archivierte PAM-Dateien.

Bezüglich der Darstellung von delta-gespeicherten PLAM-Bibliothekselementen in der Dateienliste siehe Seite 44 und 82.

Dateien / Bibliothekselemente gemäß CCS umcodieren

ONX**CONV** ['*str1*'= '*str2*',] CCS=*ccsneu* [, FCCS=*ccsalt*]

Die Sätze der durch den Action-Code X gekennzeichneten Dateien bzw. PLAM-Bibliothekselemente werden gemäß dem angegebenen Coded Character Set Namen umcodiert.

CCS=*ccsneu* im System gültiger Coded Character Set Name.
Der Inhalt der Datenobjekte wird mittels XHCS auf den angegebenen CCS umcodiert.
Falls eine Datei im Katalog bereits mit einen CCS *ccsx* verknüpft ist, wird angenommen, daß die Sätze in diesem CCS vorliegen und es erfolgt eine Umcodierung in der Form *ccsx* --> *ccsneu*

FCCS=*ccsalt* Falls die Datei im Katalog mit keinem CCS verknüpft ist, wird mit FCCS angegeben, in welcher Codierung die Eingabesätze vorliegen. In diesem Fall erfolgt die Umcodierung in der Form *ccsalt* --> *ccsneu*

Hinweis:

Ab OSD V7.0 können Dateiinhalte auch in Unicode (UTF16/UTF8/UTFE), bzw. zwischen den einzelnen Unicode-Varianten umcodiert werden.

Kopieren von Datenobjekten mit Austausch eines Stringmusters im Namen

ONX**COPY** ['*str1*'= ... [, *vsn*/*device* | [:*x*:] [,*smspar*]]] [, STD | KD]

'*str1*'= ... Namenstransformationsregeln

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

'*str1*'='*str2*'
Die Namen der kopierten Datenobjekte werden gebildet, indem in den ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und durch '*str2*' ersetzt wird. Definitionsgemäß wird der String '␣' am Ende und der Nullstring '' am Anfang eines jeden Namens gefunden.

'*str1*'=DP[*n*][I]
Delete Prefix: Der Name des neuen Datenobjekts wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der links davorstehende Namensteil bzw. die *n* links davorstehenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht.

'*str1*'=DS[*n*][I]
Delete Suffix: Der Name des neuen Datenobjekts wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der nachfolgende Namensteil bzw. die *n* nachfolgenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht. Näheres siehe Seite 109.

Enthält der ursprüngliche Name die Zeichenfolge '*str1*' nicht, so wird die COPY-Operation nicht durchgeführt.

Fehlt die Angabe '*str1*'=..., so wird "=" ergänzt (siehe unten, Beispiel 6).

vsn/device Die kopierten Dateien werden auf der angegebenen Privatplatte eingerichtet.

:x: Die kopierten Dateien werden in dem angegebenen PVS eingerichtet.

smspar Falls die kopierte Datei auf einem SM-Pubset (System Managed Pubset) angelegt wird, so kann mit diesem Parameter in Abhängigkeit von den Vorgaben des Systemverwalters der Ablageort (Volume-Set) der Datei genauer bestimmt werden.
Falls vor *smspar* der Parameter *:x:* nicht angegeben wurde, so bezieht sich *smspar* auf den Default-Pubset der Benutzerkennung.

AVAILABILITY=HIGH
Die Datei soll innerhalb eines SM-Pubsets auf einem Volume-Set abgelegt werden, der erhöhte Ausfallsicherheit besitzt (z.B. durch DRV-gespiegelte Platten).

STORAGE-CL=*class*
Name einer vom Systemverwalter eingerichteten Storage Klasse. Über die Storage Klasse wird das Volume-Set der Datei und damit die Speichereigenschaften wie z.B. Performance Attribute, Availability festgelegt.

VOLUME-SET=*vset*
Bestimmt das Volume-Set, auf dem die Datei innerhalb des SM-Pubsets eingerichtet werden soll. Der Systemverwalter kann mit *CONTROL bzw. mit der Angabe einer ein- bis vierstelligen Cat-Id *vset* (ohne ':') einen Volume-Set explizit angeben.
Der nichtprivilegierte Benutzer kann einen Volume-Set nur dann angeben, wenn ihm der Systemverwalter das Recht zur physikalischen Allokierung gegeben hat.

WORK-FILE=YES|NO
Gibt an, ob die Datei eine Arbeitsdatei ist, die von der Systemverwaltung zu einem bestimmten Zeitpunkt gelöscht werden darf. Work-Dateien werden auf einem Volume-Set abgelegt, für das der Systemverwalter die Nutzungsart WORK vorgegeben hat.

STD Dieser Parameter ist nur beim Kopieren von Dateien oder Jobvariablen von Bedeutung. Standardmäßig werden von der Variablen Action ONXCOPY die Schutzattribute der Quelldateien wie z.B. ACCESS=READ, RETPD oder Passworte auf die Zieldateien übertragen (COPY ..,SAME).

Der Operand STD bewirkt, daß die Quelldateien nicht mit "COPY ...,SAME" kopiert werden. Die Schutzattribute der Quelldateien werden somit nicht automatisch auf die Zieldateien übertragen.

KD AB OSD V8.0: Das Change-Date der Originaldatei wird auf die kopierte Datei übertragen.

Hinweise:

Bibliothekselemente werden unter Namensänderung in der aktuellen Bibliothek kopiert. Intern wird diese Variable Action umgesetzt in ONXADD '*str1*'='*str2*' *bibl*. Die Variable Action ONXCOPY ist nur bei PLAM-Bibliotheken anwendbar.

Im Unterschied zum Copy-Kommando des BS2000 werden bei der Variablen Action ONXCOPY die Attribute der Ursprungsdateien übernommen (**COPY ,SAME**). Die erzeugten Kopien bzw. die durch ONXCOPY überschriebenen Dateien erhalten die gleichen Share-/Access-Attribute/Backup-Level wie die Ursprungsdateien.

Beim Kopieren von Dateien mit BLKCTRL=PAMKEY auf ein Nonkey PVS wird von der Variablen Action intern das Programm PAMCONV aufgerufen.

Kopierte Dateien werden automatisch am Ende der aktuellen Dateienliste ergänzt.

Beispiele:

`ONXCOPY''='ABC.S.'` Voranstellen des Prefix 'ABC.S.'
z.B. TEST --> ABC.S.TEST

`ONXCOPY' '='.88'` Nachstellen des Suffix '.88'
z.B. ABC.S.TEST --> ABC.S.TEST.88

`ONXCOPY' '='.88',:A100:` Nachstellen des Suffix '.88'
z.B. [*:xxx:*]ABC.S.TEST --> :A100:ABC.S.TEST.88

`ONXCOPY' '=DP7`
Die jeweils letzten sieben Stellen im Dateinamen werden entfernt.
z.B. ABC.S.TEST.950510 --> ABC.S.TEST

`ONXCOPY''=''`
Diese Variable Action ist nur sinnvoll, falls die selektierte User-Id nicht die eigene ist. Die Dateien/Jobvariablen der selektierten Benutzerkennung werden unter dem gleichen Namen in die eigene User-Id kopiert.

`ONXCOPY''='':SRC:`
`ONXCOPY:SRC:`
Die mit X markierten Dateien werden unter dem gleichen Kennung und dem gleichen Namen auf den Pubset :SRC: kopiert.
z.B. ABC.S.TEST.50510 --> :SRC:$userid.ABC.S.TEST.950510

Löschen von Bibliothekselementen

ONX**DEL** Die mit X angekreuzten Bibliothekselemente werden gelöscht. Bei deltagespeicherten PLAM-Elementen wird nur die in der Liste ausgewiesene Version gelöscht.

Datenobjekte mit DO-Prozedur in Connection bearbeiten

ONX**DO** [*n*] ['*str1*'= ...] { *datei* | *bibliothek*(*element*) | %*name* } [, *do-params* | ?]

Die angekreuzten Datenobjekte werden mit einer DO-Prozedur bearbeitet. Dies geschieht in der Weise, daß der Name des angekreuzten Datenobjekts als erster Stellungsparameter an die DO-Prozedur übergeben wird. Es wird eine Hauptprozedur mit CALL-Aufrufen auf die Verarbeitungsprozedur erstellt. Anschließend wird in die Connection *n* verzweigt und ein DO-Aufruf für die Hauptprozedur abgesetzt.

n Nummer der Connection, in der die erzeugte Prozedur ablaufen soll. Die Connection mit der angegebenen Nummer muß bereits eröffnet sein und zwar unter der gleichen Kennung wie der CFS-Grundprozeß. Wurde keine Connection-Nummer angegeben, so wird die DO-Prozedur erzeugt, aber nicht ausgeführt.

'*str1*'= ... Namenstransformationsregeln. Näheres siehe Seite 109.

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

Der an die Verarbeitungsprozedur übergebene Name wird in der Weise gebildet, daß '*str1*' im Originalnamen durch '*str2*' ersetzt wird bzw. die davor oder dahinterliegenden

Zeichen gelöscht werden. Enthält der Originalname die Zeichenfolge '*str1*' nicht, so wird dieser Name unverändert an die Prozedur übergeben.

Ohne Angabe von '*str1*'= ... wird der Verarbeitungsprozedur der Name des zu bearbeitenden Datenobjekts in folgender Form übergeben:

- Dateien/Jobvariablen/FMS-Bibliothekselemente:
  vollständiger, bis zu 44 Byte langer Name des Datenobjekts

- PLAM-Bibliothekselemente:
  Name ohne Typ- und Versionsbezeichnung

- Elemente einer LMS-ISAM Bibliothek:
  die ersten acht Stellen des Namens

Beispiele:

...`'CFS.S.'=''`...	Das Prefix 'CFS.S.' wird aus dem Namen entfernt
...`''='CFS.S.'`...	Dem Namen wird das Prefix 'CFS.S.' vorangestellt
...`'.SRC '=''`...	Das Suffix '.SRC' wird aus dem Namen entfernt
...`' '='.SRC'`...	Dem Namen wird das Suffix '.SRC' angehängt

datei Name der DO-Prozedur zur Verarbeitung der angekreuzten Datenobjekte.

bibl(*element*) Die DO-Prozedur ist in dem angegebenen Bibliothekselement gespeichert. Bei PLAM-Bibliotheken muß das Element unter dem Typ J gespeichert sein.

%*name* Name des Parameter-Sets in der JRSAVE-Datei. Die mit dem Kommando SP %*name* gesicherte Parameter-Maske wird dem Benutzer erneut angeboten. Mit %*name* wird ein Bezug hergestellt sowohl auf eine bestimmte DO-Prozedur, als auch auf eine vordefinierte Belegung der DO-Parameter.

do-params zusätzliche Parameter zum Starten der DO-Prozedur. DO-Parameter können nicht zusammen mit %*name* angegeben werden.

? Es wird die Maske mit den Prozedurparametern angezeigt. Standardmäßig wird die Maske nur angezeigt, falls beim Aufruf der Prozedur außer dem ersten Stellungsparameter (Dateiname) noch weitere Parameter unbefriedigt sind.

Hinweise:

Die bei ONXDO angegebene DO-Prozedur muß mindestens einen Stellungsparameter besitzen. In diesem ersten Stellungsparameter wird von CFS beim Aufruf der Name des zu bearbeitenden Datenobjekts übergeben.

Werden Elemente einer Bibliothek mit ONXDO bearbeitet, so sollte die DO-Prozedur neben dem Namen des Elements noch einen weiteren Parameter für den Bibliotheksnamen enthalten. Dieser ist bei der Definition der Variablen Action in der Parameter-Maske für die DO-Prozedur anzugeben.

Durch die Variable Action ONXDO wird eine DO-Prozedur erzeugt, die die CALL-Aufrufe für die Verarbeitungsprozedur mit den angekreuzten Dateien als ersten Stellungsparameter enthält. Die DO-Prozedur erhält den Namen CFS.DO.*tsn*.*hhmmss*. *tsn* ist die TSN des Grundprozesses von CFS und *hhmmss* die aktuelle Uhrzeit. Der Benutzer ist selbst für das Löschen der Prozedur verantwortlich.

Im Zusammenhang mit der Variablen Action ONXDO sei auch auf die Variable Action ONXENTER auf Seite 133 hingewiesen.

Beispiel:

`ONXDO1 CFS.DO.UMSETZEN`
Die Prozedur CFS.DO.UMSETZEN besitzt folgende PROC-Anweisung:

```
/PROC C,(&DATEI),SUBDTA=&
```

Angenommen, der Benutzer hat Dateien seiner eigenen Benutzerkennung selektiert und durch OC1/ einen Nebenprozeß unter der eigenen Benutzerkennung eröffnet. Durch die oben angegebene Variable Action werden alle mit X in der Dateienliste markierten Dateien durch die DO-Prozedur CFS.DO.UMSETZEN bearbeitet. Dabei wird zunächst eine Hilfsprozedur mit dem Namen CFS.DO.4711. 161411 (161411 = Uhrzeit) und folgendem Inhalt erzeugt:

```
/PROC C
/CALL CFS.DO.UMSETZEN,(name-1,)
/STEP
/CALL CFS.DO.UMSETZEN,(name-2,)
/STEP
.....

/ENDP
```

Nach Erstellung der Prozedur verzweigt CFS in die bei ONXDO angegebene Connection, in unserem Beispiel in die Connection 1 und setzt dort das BS2000-Kommando /DO CFS.DO.4711.161411 ab. Nach Beendigung muß der Benutzer die Prozedur löschen.

DO-Prozeduren generieren

ONX**DPF** *ausdat* , *param*

ausdat DPF erzeugt als Ausgabedatei eine DO-Prozedur mit folgender Struktur:

```
/PROC A
expand-1
/STEP
expand-2
/STEP
....

/ENDP
```

param Die Sätze *expand-x* (*x* = 1, 2, ...) werden nach folgender Regel aus der *param*-Angabe in der DPF-Anweisung generiert: Der Text von *param* wird nach *expand* übertragen. Wo immer ein Stern (*) in *param* vorkommt, wird der mit X angekreuzte Dateiname eingesetzt. Nach diesem Schema wird für jede markierte Datei ein expandierter Satz in der Ausgabedatei von DPF erzeugt, bei dem * jeweils durch den aktuellen Dateinamen substituiert ist. Falls *param* nicht mit '/' beginnt, werden alle Leerzeichen entfernt. *param* sollte daher stets mit führendem Schrägstrich angegeben werden (/*param*).

Beispiele:

`ONXDPF DO.PROC,/CALL *,PAR=TEST`
Die Datei DO.PROC wird mit folgendem Inhalt erstellt:

```
/PROC A
/CALL dateiname-1,PAR=TEST
/STEP
/CALL dateiname-2,PAR=TEST
/STEP
....

/ENDP
```

`ONXDPF DO.PROC-A,/CALL DO.TRANSFER,*`
Die Datei DO.PROC-A wird mit folgendem Inhalt erstellt:

```
/PROC A
/CALL DO.TRANSFER,dateiname-1
/STEP
/CALL DO.TRANSFER,dateiname-2
/STEP
....

/ENDP
```

`ON&DPF DO.PROC-B,/CATJV *,*.ALT,STATE=U`
Die Datei DO.PROC-B wird mit folgendem Inhalt erstellt:

```
/PROC A
/CATJV jobvar-1,jobvar-1.ALT,STATE=U
/STEP
/CATJV jobvar-2,jobvar-2.ALT,STATE=U
/STEP
....

/ENDP
```

Dateinamen wegschreiben

ONX**DPF** *ausdat* Bei Weglassung des Operanden *param* wird für jede durch X angekreuzte Datei/Jobvariable/Bibliothekselement lediglich ein Namenssatz der Form

```
$user-id.dateiname             ,dateiname
<----------- 54 Stellen ------> <-- 44 Stellen --->
```

ausdat in die angegebene Ausgabedatei geschrieben. Die Datei enthält keine weiteren Sätze. Es fehlen insbesondere die Prozeduranweisungen /PROC, /STEP und /ENDP.

Die Ausgabedatei mit den darin enthaltenen Datei-/Elementnamen kann im EDT weiterbearbeitet werden, z.B. durch Voranstellen eines Prefix (Kommandoname), Anfügen eines Suffix (Kommando-Operanden) usw. Durch Einfügen einer PROC- und ENDP-Anweisung kann schließlich aus dem Rohgerüst der Dateinamen auf einfache Weise eine DO-Prozedur erzeugt werden.

Datenobjekte mit EDT-Prozedur bearbeiten

ONX**EDT** [*d*] [,*p*,] [*datei* | *bibliothek* ([*t*/] *element*) [, D]]

Es werden die angekreuzten Datenobjekte mit der angegebenen EDT-Prozedur (Input-Datei) bearbeitet.

d EDT-Ebene, in die die Datenobjekte eingelesen werden. Standard: 0

p EDT-Ebene, in die die Input-Prozedur eingelesen wird. Standard: 1

datei Datei, in der die Input-Datei zur Bearbeitung der Datenobjekte gespeichert ist.

bibliothek ([*t*/] *element*)
Name der Bibliothek und des Elements, in dem die Input-Datei zur Bearbeitung der Datenobjekte enthalten ist. Bei einem Element aus einer PLAM-Bibliothek kann der Elementtyp weggelassen werden. Es wird dann der Standardtyp S/ angenommen.

Die in Form einer Datei bzw. als Bibliothekselement angegebene EDT-Prozedur wird mit @INPUT in die EDT-Ebene 1 eingelesen. Die in der Prozedur enthaltenen EDT-Anweisungen müssen mit zwei @-Zeichen beginnen. Beispiel: @@ON&C'...'T'...'

D Der Parameter D bewirkt, daß nach Abarbeitung der EDT-Prozedur, mit dem EDT-Kommando @DIALOG auf Benutzerdialog im EDT umgeschaltet wird. Durch Drücken der K1-Taste kann der Benutzer den Dialog beenden. Bezüglich des Zurückschreibens oder Nicht-Zurückschreibens der virtuellen EDT-Datei wird auf den folgenden Absatz verwiesen.

Für jedes Datenobjekt wird die Bearbeitung in folgenden Schritten ausgeführt:

a) Das Datenobjekt wird in die EDT-Ebene *d* (Standard: 0) eingelesen.

b) Der Prozeßschalter 0 wird zurückgesetzt (/SETSW OFF=0).

c) Die von CFS mit @INPUT in die EDT-Ebene *p* (Standard: 1) eingelesene EDT-Prozedur wird mit dem Kommando @DO *p* zur Ausführung gebracht.

d) Nach Abarbeitung der EDT-Prozedur bzw. nach Beendigung des Benutzerdialogs durch Drücken der K1-Taste wird das bearbeitete Datenobjekt von CFS automatisch zurückgeschrieben, sofern nicht eine der folgenden Bedingungen erfüllt ist:

- In der EDT-Prozedur oder durch den Benutzer wurde der Prozeßschalter 0 gesetzt (@SETSW ON=0).

- In der EDT-Prozedur oder durch den Benutzer wurde die virtuelle EDT-Ebene *d* gelöscht (@DEL).

Hinweise:

Die Variable Action ONXEDT steht ab EDT V16.2 zur Verfügung.

Bei Angabe der Variablen Action ONXEDT ohne Parameter wird keine EDT-Prozedur gestartet. Es wird sofort in den Benutzerdialog zur individuellen Bearbeitung des Datenobjekts verzweigt. Ein Update des Datenobjekts erfolgt nicht, falls vor der Rückkehr

nach CFS (K1-Taste) die EDT-Ebene 0 durch @DEL gelöscht wurde oder wenn der Prozeßschalter 0 gesetzt ist @SETSW ON=0.

In der Input-Datei können weitere EDT-Prozedurebenen definiert werden (@PROC *n* ... @END).

Nach dem erfolgreichem Einlesen eines Datenobjekts in den Datenbereich des EDT wird von CFS eine SDF-P Variable mit dem Namen CFSEDT mit folgendem Inhalt gefüllt:

Spalte	Länge	Inhalt
1	54	Name der eingelesenen Datei/des eingelesenen Bibliothekselements PLAM: Name ohne Typ und Versionsbezeichnung
55	8	PLAM: Elementtyp, z.B. 'S'
63	24	PLAM: Elementversion
87	54	bei Bibliothekselementen: Name der Bibliothek
141	1	Art des eingelesenen Datenobjekts: C'⊔' Datei C'J' Jobvariable C'P' PLAM-Element C'L' LMS-ISAM-Element C'F' FMS-Element

Beispiel:

Die unter dem Namen ASS.EDT gespeicherte Input-Datei zur Bearbeitung von Assembler-Quellprogrammen enthält z.B. folgende Anweisungen:

```
@@DEL&:73-80:
@@ON&CA'=''  'T'=C''  '
```

Durch die Variable Action `ONXEDT ASS.EDT` werden bei allen mit dem Action-Code X markierten Datenobjekten die Spalten 73 bis 80 gelöscht. Aufgrund der zweiten Anweisung werden alle Zeichenfolgen "='⊔" in "=C'⊔" umgesetzt.

Datenobjekte mit DO-Prozedur in Enter bearbeiten

ONX**ENTER** ['*str1*'=...] {*datei*|*bibl*(*element*)|%*name*|%?} [,*do-params*|,?] [/[*enter-param*] [,JRF=*jr-file*] [, JN=*jobname*] [, EM]]

Die angekreuzten Datenobjekte werden mit einer DO-Prozedur bearbeitet. Dies geschieht in der Weise, daß der Name des angekreuzten Datenobjekts als erster Stellungsparameter an die DO-Prozedur übergeben wird. Es wird eine Hauptprozedur mit CALL-Aufrufen auf die Verarbeitungsprozedur erstellt. Die DO-Prozedur wird anschließend durch das Job-Report System im Enter gestartet. Die Beendigung des Enter-Jobs wird überwacht und dem Benutzer in Form der Job-Report Maske mitgeteilt. Für nähere Informationen zum Job-Report System siehe Seite **397**.

'*str1*'= ... Namenstransformationsregeln. Näheres siehe Seite **109**.

'*str1*'='*str2*' | '*str1*'=DP[*n*][l] | '*str1*'=DS[*n*][l]

Der an die Verarbeitungsprozedur übergebene Name wird in der Weise gebildet, daß '*str1*' im Originalnamen durch '*str2*' ersetzt wird bzw. die davor oder dahinterliegenden Zeichen gelöscht werden. Enthält der Originalname die Zeichenfolge '*str1*' nicht, so wird dieser Name unverändert an die Prozedur übergeben.

Ohne Angabe von '*str1*'=... und falls *do-params* für die Verarbeitungsprozedur keine der Platzhalter-Zeichenfolgen !NAME, !ELEMENT, !LIBRARY, !TYP, !VERS enthält, wird der Verarbeitungsprozedur der Name des zu bearbeitenden Datenobjekts in folgender Form übergeben:

- Dateien/Jobvariablen/FMS-Bibliothekselemente:
  vollständiger, bis zu 44 Byte langer Name des Datenobjekts

- PLAM-Bibliothekselemente:
  Name ohne Typ- und Versionsbezeichnung

- Elemente einer LMS-ISAM Bibliothek:
  die ersten acht Stellen des Namens

Falls '*str1*'=... nicht angegeben ist und *do-params* eine der Platzhalter-Zeichenfolgen !NAME, !ELEMENT, !LIBRARY, !TYP oder !VERS enthält, so wird der Verarbeitungsprozedur von CFS nicht als erster Stellungsparameter ein Name übergeben. Es werden vielmehr in den *do-params* die Platzhalter-Zeichenfolgen mit folgenden Inhalten substituiert:

!NAME	Name der Datei/Jobvariablen
!ELEMENT	Bezeichnung des Bibliothekselements ohne Typ und Version
!LIBRARY	Name der Bibliothek, in der das Element enthalten ist
!TYP	Typ des Bibliothekselements
!VERS	Version des Bibliothekselements

Beispiele:

...`'CFS.S.'=''`...	Das Prefix 'CFS.S.' wird aus dem Namen entfernt
...`''='CFS.S.'`...	Dem Namen wird das Prefix 'CFS.S.' vorangestellt
...`'.SRC '=''`...	Das Suffix '.SRC' wird aus dem Namen entfernt
...`' '='.SRC'`...	Dem Namen wird das Suffix '.SRC' angehängt

datei Name der Verarbeitungsprozedur.

bibl(*element*) Die DO-Prozedur ist in dem angegebenen Bibliothekselement gespeichert. Bei PLAM-Bibliotheken muß das Element unter dem Elementtyp J gespeichert sein.

%*name* Name des Parameter-Sets in der JRSAVE-Datei. Die mit dem Kommando SP %*name* gesicherte Parameter-Maske wird dem Benutzer erneut angeboten. Mit %*name* wird ein Bezug sowohl auf eine bestimmte DO-Prozedur, als auch auf eine vordefinierte Belegung der DO-Parameter hergestellt hergestellt.

%? Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

do-params, *enter-params*, *jr-file*, *jobname*

zusätzliche Parameter zum Starten der DO-Prozedur im Enter. DO-Parameter können nicht zusammen mit %*name* angegeben werden. Für eine ausführliche Beschreibung der einzelnen Parameter wird auf Seite **397** und folgende (Job-Report System) verwiesen.

EM Enter Multiple. Für jeden DO-Aufruf wird ein eigener Enter-Auftrag erzeugt und als solcher getrennt von allen anderen durch den Job-Report überwacht.

Standard: Es wird nur ein Enter-Auftrag gestartet. Innerhalb des Enters wird für jede angekreuzte Datei ein /CALL *do-prozedur*,dateiname ausgeführt.

Hinweise:

Die bei ONXENTER angegebene DO-Prozedur muß mindestens einen Stellungsparameter besitzen. In diesem ersten Stellungsparameter wird ein 44-stelliger Datei / Elementname übergeben. Die DO-Prozedur sollte funktionell so aufgebaut sein, daß in ihr eine Datei / ein Bibliothekselement, dessen Name im ersten Stellungsparameter zu übergeben ist, in einer bestimmten Weise bearbeitet wird.

Werden Elemente einer Bibliothek mit ONXENTER bearbeitet, so sollte die DO-Prozedur neben dem Namen des Elements mindestens noch einen Parameter für den Bibliotheksnamen enthalten. Dieser ist bei der Definition der Variablen Action in der Parameter-Maske für die DO-Prozedur anzugeben.

Beim Starten einer DO-Prozedur im Enter und JRF=SYSLST wird in der Enter-Datei neben dem Kommando /SYSFILE SYSLST auch /OPTION MSG=FH eingefügt. Damit wird erreicht, daß auch die SYSOUT-Ausgaben in der SYSLST-Datei enthalten sind.

Im Zusammenhang mit der Variablen Action ONXENTER sei auch auf die Variable Action ONXDO auf Seite 128 hingewiesen.

Beispiele:

```
ONXENTER CFS.S.LMSLIB(ASSEMB),LIB=CFS.SRCLIB/
JOB-CLASS=JCBLONG,TIME=500
```

Die Prozedur ASSEMB in der Bibliothek CFS.S.LMSLIB hat folgende PROC-Anweisung:

```
/PROC N,(&PROGNAM,                  ":Name des Quellprogramms"    -
/        &LIB=,                     ":Name der Source-Bibliothek"  -
/        &IDA=NO,                                                 -
/        &XREF=NOXREF":XREF/NOXREF/ATXREF/NOATXREF" -
/        ),SUBDTA=&
```

Angenommen, der Benutzer hat durch NPLMS CFS.SRCLIB alle Quellprogramme in der LMS-Bibliothek CFS.SRCLIB selektiert. Durch die oben angegebene Variable Action werden alle mit X markierten Quellprogramme mit der Prozedur ASSEMB übersetzt. Es wird hierzu die folgende Enter-Datei erzeugt:

```
/LOGON
/SYSFILE SYSOUT=CFS.SYSOUT.tsn.time
/STEP
/DO CFS.S.LMSLIB(ASSEMB),(name-1,LIB=CFS.SRCLIB,IDA=NO,XREF=NOXREF,)
/STEP
/DO CFS.S.LMSLIB(ASSEMB),(name-2,LIB=CFS.SRCLIB,IDA=NO,XREF=NOXREF,)
/STEP
.....

/LOGOFF NOSPOOL
```

Die Enter-Datei wird über das Job-Report System von CFS gestartet. Als Enter-Parameter werden verwendet JOB-CLASS=JCBLONG und TIME=500. Nach Beendigung des Enter-Auftrags wird dem Benutzer das Ablaufprotokoll des Jobs in Form des SYSOUT-Listings CFS.SYSOUT.*tsn.time* präsentiert.

`ONXENTER CFS.S.LMSLIB(ASSEMB),LIB=CFS.SRCLIB/EM`
Aufgrund der EM-Option (Enter Multiple) wird für jedes angekreuzte Source Element eine eigene Enter-Datei erzeugt, z.B.

```
/LOGON
/SYSFILE SYSOUT=CFS.SYSOUT.tsn.time
/STEP
/DO CFS.S.LMSLIB(ASSEMB),(name-1,LIB=CFS.SRCLIB,XREF=NOXREF,)
/STEP
/LOGOFF NOSPOOL
```

Jede der so erzeugten Enter-Dateien wird separat über das Job-Report System gestartet. Über die Beendigung jedes einzelnen Enter-Auftrags wird der Benutzer durch Ausgabe des SYSOUT-Protokolls informiert.

Datenobjekte nach Zeichenfolgen durchsuchen

ONX**FIND** [*n*,] *param* [=W *datei* [, E|O]] | =P] [, NO] [, C[O]] [, +-*n*|+*n*|-*m*] [, NC] [, F] [, PD] [,SKIP[F]] [,TAPE] [,TYPE=*t1*,*t2*,...]

Durchsuchen aller angekreuzten Datenobjekte (Dateien/Bibliothekselemente/Jobvariablen) nach dem Vorkommen eines oder mehrerer Suchbegriffe in den Daten bzw. im Namen. Am Bildschirm wird die Anzahl der Sätze ausgegeben, in denen der/die Suchbegriffe gefunden wurden. Die Trefferanzahl wird außerdem rechtsbündig im Action-Code Feld des betreffenden Datenobjekts eingetragen, falls die Terminierungsabfrage mit U (Updated List) beantwortet wurde.

Mit dem Zusatz =W *datei* wird zusätzlich eine druckaufbereitete Liste erzeugt. Diese enthält die Namen der Datenobjekte, die Treffer gebracht haben. Die Treffersätze werden mit ihrem gesamten Inhalt daran anschließend in der Write-Datei aufgelistet. Wahlweise kann auch das Umfeld der Treffer, d.h. die Sätze davor und danach ausgegeben werden.

n Beschränkung der Suche auf die ersten *n* Sätze. (Standard: alle Sätze).

param einfache oder mehrfache Suchanweisung.

einfache Suchanweisung: [*col*] [*p*] *item*

col Spaltenbereich in dem die gesuchte Zeichenfolge beginnen muß.

:col1-col2: Das erste Zeichen der gesuchten Zeichenfolge muß im Spaltenbereich zwischen col1 und col2 beginnen.

:col1: Die Zeichenfolge wird nur an der angegebenen Spalte col1 gesucht und muß dort beginnen.

>:col1: | <:col1: Die Zeichenfolge wird im Bereich ab Spalte col1 bis Satzende (>) bzw. vom Satzanfang bis Spalte col1 gesucht (<)

Standard: Die Suche erstreckt sich von Spalte 1 eines jeden Satzes bis zum jeweiligen Satzende.

p > Suche nach einer Zeichenfolge > *item*
< Suche nach einer Zeichenfolge < *item*

- Suche nach einer Zeichenfolge ungleich *item*

Standard: Suche nach einer Zeichenfolge = *item*

item Suchzeichenfolge: C'*string*' | L'*string*' | X'*string*' | A'*string*'
C'*string*' kann zu '*string*' abgekürzt werden.
L'*string*': Kleinbuchstaben in *string* werden nicht in Großbuchstaben umgesetzt.
A'*string*': Bei der Suche werden Klein- und Großbuchstaben gleich behandelt. Beispiel: Der Suchbegriff A'CFS' wird als Treffer erkannt, falls im Datensatz die Zeichenfolge 'CFS' oder 'cfs' oder eine beliebige Kombination aus Groß- und Kleinbuchstaben enthalten ist.

Enthält *string* Hochkommas, so müssen diese verdoppelt angegeben werden ('').

Mehrfachsuche - Format 1: *param* [*vk param*] [*vk param*] .....

param einfaches Suchargument gemäß der oben beschriebenen Syntax.

vk Verknüpfung mit dem vorausgegangenen einfachen Suchargument.

, Oder-Verknüpfung.

+ Und-Verknüpfung.

***** [*n*] Wildcard-Verknüpfung: Und-Verknüpfung, jedoch muß das zweite Suchitem im Datensatz nach dem ersten Suchitem vorkommen. Der optionale Zusatz *n* legt die Anzahl der zwischen den beiden Suchitems zu stehenden Trennzeichen fest.

Es können beliebig viele Konstrukte der Art *vk such* aneinandergereiht werden.
Bei Verknüpfung mehrerer Und-Bedingungen können auch geklammerte Ausdrücke angegeben werden: (.. , .. , ...)+(.. , .. , ...). Ein Beispiel hierzu ist auf Seite **315** beschrieben.

Hinweis:

Die Und-/Oder-Verknüpfung ist jeweils auf einen Datensatz bezogen. Dies bedeutet, daß beide Suchbegriffe im selben Satz enthalten sein müssen. Für eine ausführliche Beschreibung der Verknüpfungsoperationen siehe Seite **313**.

Mehrfachsuche - Format 2: ( )

Durch () wird eine spezielle Such-Maske angefordert. Dort stehen dem Benutzer 18 Bildschirmzeilen zur Aufnahme von umfangreichen Suchargumenten zur Verfügung. Für eine Beschreibung der Such-Maske siehe Seite **315**.

Mehrfachsuche - Format 3: %*name* | %? [, *jrsave*]

%*name* ist die Bezeichnung eines Parameter-Sets, unter dem zu einem früheren Zeitpunkt ein komplexes Suchargument gespeichert wurde (Kommando SP %*name*). Näheres hierzu siehe Seite **315**.
%?: Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Markieren mit 'x' kann ein Parameterset ausgewählt werden. Durch Markieren

mit 'm' (Modify) wird der Inhalt des Parametersets angezeigt und kann noch verändert werden.
jrsave ist der Name der Datei, in der der Parameter-Set gespeichert ist. Standard: CFS.JRSAVE [*.jobname*] .

Mehrfachsuche - Format 4: (*s-dat*)

s-dat ist der Name einer Datei (SAM), in der die Suchargumente festgehalten sind. Näheres hierzu siehe Seite 316.

Mehrfachsuche - Format 5: ? | ?,%*name*

Durch ? wird das zuletzt definierte Suchargument wieder angezeigt.
Durch ?,%*name* wird die in %*name* gespeicherte Suchbedingung angezeigt.

Wegschreiben der Treffersätze

ONXFIND ... =W *datei* [, E | O]

Die Treffersätze werden in eine druckaufbereitete Datei geschrieben. Die Namen der entsprechenden Datenobjekte werden in der Write-Datei ebenfalls dokumentiert. Bei Angabe der NO-Option (siehe unten) werden nur die Namen dokumentiert.

E — Die Write-Datei wird mit Open=Extend eröffnet.

O — Die Write-Datei wird mit Open=Output eröffnet.

Ausgabe der Treffersätze am Bildschirm

ONXFIND ... =P

Der Inhalt der Treffersätze wird am Bildschirm ausgegeben. Diese Option entspricht dem Suche-Kommando mit Direktausgabe der Treffersätze. Näheres, Seite 321.

Suche erstreckt sich auf Datenobjekt Namen und nicht auf Dateninhalte

ONXFIND ..., NO

Names Only. Es werden nur die Namen der Datenobjekte nach dem Vorkommen der angegebenen Strings durchsucht. Falls eine Bibliothek auf Dateiebene mit der NO-Option nach Namen durchsucht wird, so werden die Elemente, deren Namen den Suchstring enthalten, am Bildschirm ausgegeben (ONXFIND...=P,NO) bzw. in die Write-Datei geschrieben (ONXFIND...=W,NO).
Der Anwendungsbereich für die NO-Option liegt vor allem im Durchsuchen mehrerer Bibliotheken nach Elementen mit einem bestimmten Namen. Siehe hierzu das zweite Beispiel weiter unten.

ONXFIND ..., C

Check for Continuation (Standard): Es werden bei Prozeduren/Makros/Quellprogrammen alle Sätze als logische Einheit betrachtet, die durch Fortsetzungszeichen miteinander verbunden sind. Dies bedeutet, daß z.B. der zusammengesetzte Suchbegriff 'FCB'*'BLKCTRL=DATA' in der Datei als Treffer gefunden wird, obwohl der Teilstring 'FCB' in einer Zeile, und der Teilstring 'BLKCTRL=DATA' in einer Folgezeile enthalten ist.
Bei der Ausgabe der Treffersätze in eine Datei werden alle Sätze der logischen Einheit in die Write-Datei übertragen. Falls der Suchstring vollständig in einem der Teil-

sätze enthalten ist, so wird dieser Teilsatz in der Ausgabedatei durch das Zeichen 'X' in Spalte 2 hervorgehoben.

ONXFIND ..., CO

Check for Continuation Old mode. Falls bei Prozeduren/Makros/Quellprogrammen ein Treffer in einem Satz gefunden wurde, der mit einem Fortsetzungszeichen endet, so werden bei der Ausgabe der Treffersätze in eine Datei auch alle folgenden, mit einem Fortsetzungszeichen endenden Sätze in die Write-Datei übertragen. Um als solcher erkannt zu werden, muß der Suchstring im CO-Modus vollständig in einem der Teilsätze enthalten sein.

ONXFIND ... , +-*n* | +*n* | -*m*

Das Umfeld der Treffer wird in die Write-Datei ausgegeben.

+-*n* Es werden *n* Sätze (1 <= *n* <= 250) vor dem Treffer, der Treffer selbst und *n* Sätze nach dem Treffer in die Write-Datei geschrieben.

[+*n*] [-*m*] Es werden *m* Sätze (1 <= *m* <= 250) vor dem Treffer, der Treffer selbst und *n* Sätze (1 <= *n* <= 250) nach dem Treffer in die Write-Datei geschrieben. Es kann der Zusatz +*n* oder -*m* weggelassen werden. In diesem Fall werden nur die Sätze vor bzw. nach dem Treffer ausgegeben.

ONXFIND ..., NC

No Check for Continuation. Die Prüfung auf Fortsetzung eines BS2000-Kommandos /einer Makro-/Quellprogramm-Anweisung wird unterdrückt.

ONXFIND ..., F

Find First. Nach dem ersten gefundenen Treffer wird die Suche für das Datenobjekt beendet. Es wird dann ein Treffer gemeldet. Mit dieser Option läßt sich auf schnelle Weise feststellen, ob in einem Datenobjekt ein vorgegebener Suchbegriff enthalten ist oder nicht. Standardmäßig werden nach einem gefundenen Treffer auch alle weiteren Sätze des Datenobjekts nach dem Suchbegriff durchsucht. Die Anzahl der Treffer wird in diesem Fall mit dem korrekten Wert angezeigt.

ONXFIND ..., PD

Physical mode for archive Directories. ISAM-Dateien, die Archive-Directories darstellen, werden standardmäßig auf der logischen Ebene, d.h. auf der Ebene der gesicherten Dateinamen durchsucht. Daraus ergibt sich ein anderes Format bei der Ausgabe der Treffersätze. Der Parameter PD bewirkt, daß Archive-Directories wie normale ISAM-Dateien behandelt werden.

ONXFIND ..., SKIP[F]

SKIP Datenobjekte, bei denen der Suchbegriff nicht gefunden wurde, werden mit dem Action-Code '-' in der Dateienliste unsichtbar gemacht. Durch das Kommando YANK können diese Einträge wieder sichtbar gemacht werden.

SKIPF Datenobjekte, bei denen der Suchbegriff gefunden wurde, werden mit dem Action-Code '-' in der Dateienliste unsichtbar gemacht. Durch das Kommando YANK können diese Einträge wieder sichtbar gemacht werden.

ONXFIND ..., TAPE

Standardmäßig werden Banddateien und migrierte Dateien bei der Variablen Action ONXFIND nicht berücksichtigt. Mit dem Parameter TAPE werden auch Banddateien und migrierte Dateien durch ONXFIND bearbeitet. D.h. die entsprechenden Bänder werden angefordert und müssen montiert werden.

ONXFIND ...., TYPE=*t1*,*t2*, ...

Dieser Parameter ist nur von Bedeutung, falls Dateien durchsucht werden, hinter denen sich PLAM-Bibliotheken verbergen.
Die Suche erstreckt sich nur auf Elemente in den Bibliotheken, die vom angegebenen Typ sind. Beispiel: TYPE=J oder TYPE=J,S
Standardmäßig werden alle Elemente mit beliebigen Typ-Bezeichnungen in der Bibliothek durchsucht.

Der Zusatz =W *datei* [, E | O] kann auch in der Such-Maske (siehe: Mehrfachsuche - Format 2) bzw. in *s-dat* (siehe: Mehrfachsuche - Format 4) im Anschluß an den letzten Suchstring angegeben werden.

Hinweise:

Im Zusammenhang mit der Write-Option (=W ..) wird auf das Kommando **REWR** (Zurückschreiben der Write-Datei in die einzelnen Ursprungsdateien/ Elemente) hingewiesen. Das Rewrite-Kommando REWR (siehe Seite 271) stellt das Gegenstück zur Write-Option dar.

Sind in der Liste der zu durchsuchenden Dateien auch Bibliotheken enthalten, so durchsucht CFS die Bibliotheken auf der Ebene der Elemente. Bei Elementen mit gleichem Namen wird nur dasjenige mit der höchsten Versionsnummer durchsucht.

Beim Wegschreiben der Treffersätze aus einer BS2000-Datei unter OSD V1 oder höher bzw. aus einer PLAM-Bibliothek wird neben dem Namen des beteffenden Datenobjekts auch dessen Last Update Datum in der Trefferdatei abgelegt. Bei einem späteren Rewrite kann dann geprüft werden, ob das Datenobjekt seit dem ONXFIND-Lauf geändert wurde. Ist dies der Fall, wird der Rewrite für das Datenobjekt nicht ausgeführt, da dies inkonsistente Daten zur Folge haben kann. Über eine Modifikation im Initialisierungsmodul CFSMAIN kann das Schreiben der Last Update Sätze in die Trefferdatei aus Kompatibilitätsgründen generell verhindert werden.

Wurde hinter dem Namen der Write-Datei keine der Optionen E/O (Extend/ Overwrite) angegeben, so gilt folgende Regelung:

Falls die Write-Datei im aktuellen CFS-Lauf zum ersten Mal angesprochen wird, so wird als Open-Modus in jedem Fall O (Overwrite) angenommen, d.h. die Datei wird neu angelegt bzw. überschrieben.

Falls in mehreren Variablen Actions nacheinander die gleiche Write-Datei angegeben wurde, so wird sie standardmäßig mit Open=Extend eröffnet.

Durch die E-/O-Option kann vom Benutzer ein vom Standardfall abweichender Open-Modus angegeben werden.

Wird eine PAM-Datei, hinter der sich eine Bibliothek verbirgt, bei ONXFIND mit X markiert, so durchsucht CFS die Bibliothek auf der Ebene der Bibliothekselemente und nicht auf der elementaren Datensatz- oder Blockebene. Falls dieses Verhalten nicht gewünscht ist, kann z.B. der Parameter TYPE=Z angegeben werden.

Aufgrund des oben beschriebenen standardmäßigen Verhaltens der Variablen Action ONXFIND können durch Ankreuzen mehrerer Bibliotheken alle Elemente auf das Vorkommen bestimmter Suchbegriffe in den Daten bzw. in den Elementnamen geprüft werden. Die Variable Action FIND liefert die Gesamtanzahl der Treffersätze über alle Elemente in einer Bibliothek bzw. die Anzahl der Elementnamen, die einen der gesuchten Strings enthalten.

Beispiele:

`ONXFIND 'DEVICE=D3490'=W CFS.D349X,+-4`
In allen durch X angekreuzten Datenobjekten wird der String 'DEVICE=D3490' gesucht. Die Datensätze, in denen der Suchstring enthalten ist, werden zusammen mit dem Namen des zugehörigen Datenobjekts in die Datei CFS.D349X geschrieben. Außerdem wird das Umfeld der Treffer, d.h. 4 Sätze vor und 4 Sätze nach jedem Treffer in die Write-Datei ausgegeben.

`ONXFIND 'A1015'=W L.FNDLIST,NO`
Anforderung: Es ist ein Quellprogramm mit dem Namen A1015 in mehreren, in der Dateienliste als BS2000-Dateien dargestellten Bibliotheken zu suchen. Die in Frage kommenden Bibliotheken werden mit dem Action-Code X markiert. Die oben angegebene Variable Action durchsucht die Namen aller Elemente der markierten Bibliotheken nach dem Vorkommen des Strings A1015. In der Datei L.FND LIST werden die Elementnamen, die einen Treffer gebracht haben, zusammen mit dem entsprechenden Bibliotheksnamen aufgeführt. Nach Eingabe von U auf die Terminierungsabfrage wird in der Dateienliste im Action-Code Feld der angekreuzten Bibliotheken die Anzahl der Elemente aufgeführt, deren Namen die Suchbedingung erfüllt hat.

`ON&FIND 10 '/FILE','/SET-FILE-LINK'=W CFS.FILE.TREFFER`
Bei allen in der Dateienliste vorkommenden Datenobjekten wird in den ersten 10 Sätzen die Zeichenfolge '/FILE' oder '/SET-FILE-LINK' gesucht. Sätze, die mindestens einen Suchstring enthalten, werden nach CFS.FILE.TREFFER geschrieben.

`ONXFIND ''`
Falls diese Variable Action auf eine Datei angewendet wird, die eine Bibliothek repräsentiert, so wird als Trefferanzahl die **Anzahl der Datensätze in allen Elementen** ausgegeben, die in der Bibliothek gespeichert sind.

Bezüglich der Anzahl der Sätze eines Datenobjekts siehe auch User Option RECNT (Record-Count) Seite **99**.

`ONXFIND ''=W CFS.ALLES`
Alle Datensätze der durch X markierten Datenobjekte werden in die Datei CFS. ALLES geschrieben. Vergleiche hierzu auch die Variable Action ONXLIST.

`ON&FIND :1:'$A'=W #1`
Sucht in allen selektierten Jobvariablen (FCBTYPE=J) in Spalte 1 die Zeichenfolge '$A'. Eine Liste der Namen der entsprechenden Jobvariablen, zusammen mit ihrem Inhalt, wird in die List-Datei #1 geschrieben. Die List-Datei kann mit SPACE=E ausgedruckt werden.

Weitere Beispiele für Suchargumente, insbesondere die Verknüpfung von mehreren Suchargumenten finden Sie auf Seite **314** - **316**.

Dateien mit FLAM komprimieren

ONX**FLAM** ['*str1*'= ... [, *par*]] [, *flam-par*]

Die mit dem Action-Code X markierten Dateien werden nach der "Frankenstein-Lidzba-Access-Method" (FLAM) komprimiert. Das Ergebnis der Komprimierung ist stets eine SAM-Datei.

'*str1*'= ... Namenstransformationsregeln. Näheres siehe Seite **109**.

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

par *:x:* | *vsn/device*

:x:
Die komprimierten Dateien werden in dem Public-Volume-Set gespeichert.

vsn/device
Die komprimierten Dateien werden auf der angegebenen Privatplatte eingerichtet.

flam-par FLAM spezifische Parameter, z.B. FC=ISAM,RECFORM=VAR
Für eine ausführliche Beschreibung der FLAM-Parameter siehe FLAM-Benutzerhandbuch, Kapitel 3, "Schnittstellen". Die FLAM-Parameter können erst ab FLAM-Version 2.0 angegeben werden.

Hinweise:

Um diese Variable Action anwenden zu können, muß das Softwareprodukt FLAM verfügbar sein.

Falls keine FLAM-Parameter angegeben wurden, wird von CFS der Parameter INFO= NO ergänzt. Damit werden von FLAM lediglich Fehlermeldungen am Bildschirm protokolliert.

Bei Angabe der Variablen Action ohne '*str1*' und '*str2*', sowie ohne *vsn/device* (ONXFLAM [*flam-par*]), erhält die komprimierte Datei den gleichen Namen wie die Originaldatei. Die Originaldatei wird überschrieben.

Der Systemverwalter kann einen bestimmten BACKUP-Level als Kennzeichen für komprimierte Dateien festlegen. In diesem Fall werden die Dateien von CFS nach erfolgreicher Komprimierung auf den festgelegten BACKUP-Level gesetzt (CAT-Kommando). Komprimierte Dateien können in diesem Fall aufgrund ihres Backup-Levels leicht identifiziert werden. (CFS User Option: BACKUP).

Geflamte Dateien können im Display-Modus in entkomprimierter Form betrachtet werden, ohne daß die Datei mit DFLAM entkomprimiert werden muß. Für weitere Informationen siehe Kommando DA FLAM, Seite **302**.

Beim Entkomprimieren der Dateien (DFLAM) wird der BACKUP-Level wieder auf den im BS2000 gültigen Standardwert zurückgesetzt.

Dateien mit FLAM dekomprimieren

ONX**DFLAM** ['*str1*'= ... [, *par*]] [, *flam-par*]

Die mit dem Action-Code X markierten Dateien werden nach der "Frankenstein-Lidzba-Access-Method" (FLAM) dekomprimiert. Die zu einem früheren Zeitpunkt mit ONXFLAM komprimierte Datei wird mit allen Originalattributen wie Recform, Recsize, Blksize, Keylen, Keypos wieder hergestellt. Die Attribute Share-, Access-, Audit-, usw. werden nicht gespeichert und können somit nicht wiederhergestellt werden.

'*str1*'= ... Namenstransformationsregeln. Näheres siehe Seite **109**.

'*str1*'='*str2*' | '*str1*'=DP[*n*][l] | '*str1*'=DS[*n*][l]

par *:x:* | *vsn/device*

:x:
Die entkomprimierten Dateien werden in dem Public-Volume-Set gespeichert.

vsn/device
Die entkomprimierten Dateien werden auf der Privatplatte eingerichtet.

flam-par FLAM spezifische Parameter (siehe Benutzerhandbuch "FLAM Version 2.0"). Diese Parameter können unter Version 1 des Softwareprodukts FLAM nicht angegeben werden.

Hinweise:

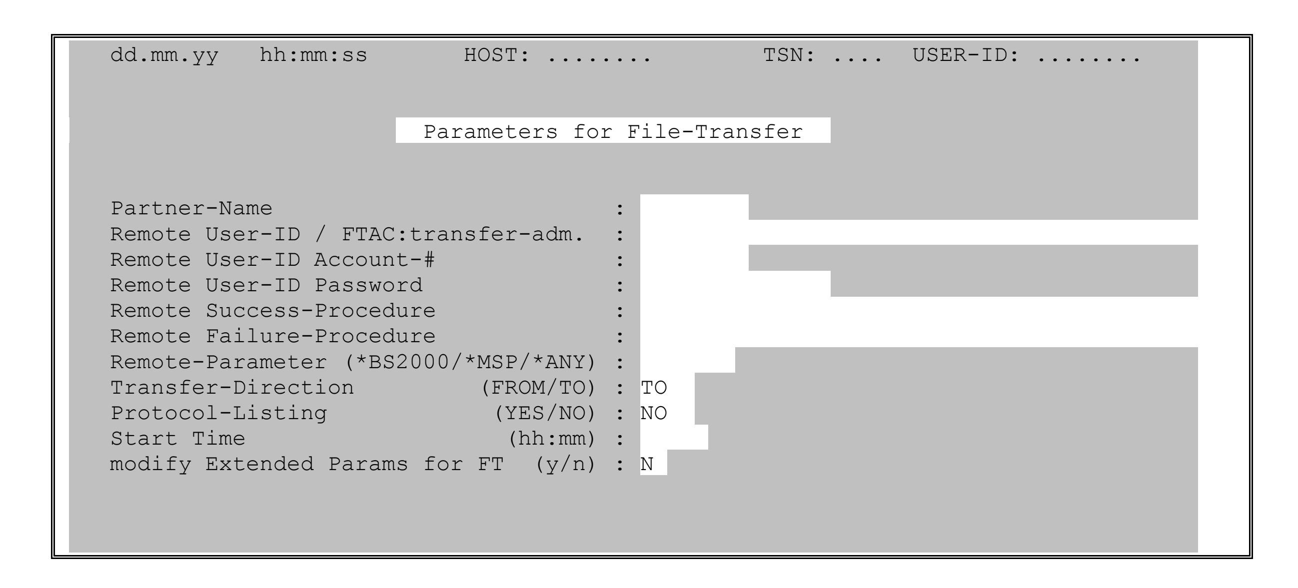
Um diese Variable Action anwenden zu können, muß das Softwareprodukt FLAM verfügbar sein.

Falls keine FLAM-Parameter angegeben wurden, wird von CFS der Parameter INFO= NO ergänzt. Damit werden von FLAM lediglich Fehlermeldungen am Bildschirm protokolliert.

Bei Angabe der Variablen Action ohne Parameter '*str1*' und '*str2*', sowie ohne *vsn/device* (ONXDFLAM [*flam-par*]), wird die komprimierte Datei durch die entkomprimierte Datei überschrieben.

Geflamte Dateien können im Display-Modus in entkomprimierter Form betrachtet werden, ohne daß die Datei mit DFLAM entkomprimiert werden muß. Für weitere Informationen siehe Kommando DA FLAM, Seite 302.

Dateien / Bibliothekselemente mit File-Transfer übertragen


```
dd.mm.yy   hh:mm:ss       HOST: ........        TSN: ....  USER-ID: ........

                        Parameters for File-Transfer

Partner-Name                                :
Remote User-ID / FTAC:transfer-adm.         :
Remote User-ID Account-#                    :
Remote User-ID Password                     :
Remote Success-Procedure                    :
Remote Failure-Procedure                    :
Remote-Parameter  (*BS2000/*MSP/*ANY)       :
Transfer-Direction           (FROM/TO)      : TO
Protocol-Listing              (YES/NO)      : NO
Start Time                     (hh:mm)      :
modify Extended Params for FT    (y/n)      : N
```

ONX**FT** ['*str1*'= ...] [%*name* | %? [,JRS=*jrfile*|,?] [,]] [*host2*] [,*user-id2*] [,*acct-nr2*] [,*passw2*] [,*rem-succ*] [,*rem-fail*] [,*rem-par*] [,*direct*] [,*lst*] [,*time*]

Falls keine Operanden angegeben sind, werden sie in einer gesonderten File-Transfer Parameter Maske angefordert (siehe oben).

'*str1*'= ... Namenstransformationsregeln

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

'*str1*'='*str2*'
Der Name der Datei im Zielsystem wird gebildet, indem im Namen der Ursprungsdatei '*str1*' gesucht und durch '*str2*' ersetzt wird. Definitionsgemäß wird der String '⊔' am Ende und der Nullstring '' am Anfang eines jeden Namens gefunden.

'*str1*'=DP[*n*][I]
Delete Prefix: Der Name der Datei im Zielsystem wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der links davorstehende Namensteil bzw. die *n* links davorstehenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht.

'*str1*'=DS[*n*][I]
Delete Suffix: Der Name der Datei im Zielsystem wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der nachfolgende Namensteil bzw. die *n* nachfolgenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht. Näheres siehe Seite **109**.

Enthält der Name der Ursprungsdatei die Zeichenfolge '*str1*' nicht, so erfolgt kein File-Transfer.

Fehlt die Angabe '*str1*'=..., so erhalten die Dateien im Zielrechner die gleichen Namen wie im Ursprungssystem.

%*name* Die Parameterangaben zum File-Transfer werden dem angegebenen Parameterset entnommen.

%? Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

JRS=*jrfile* Name der JRSAVE-Datei, in der der Parameterset gespeichert ist.
Standard: CFS.JRSAVE.*xxxx*. Näheres hierzu siehe Seite **565**.

? Die FT-Maske wird mit dem Inhalt des Parametersets gefüllt und dem Benutzer angezeigt. Standardmäßig wird die FT-Maske nicht angezeigt.

Hinweise:

Die durch den Parameterset aktivierten File-Transfer Parameter können durch nachfolgend angegebene Parameterwerte wie *host2*, *user-id2* modifiziert werden.

Das Speichern der zuletzt benutzten FT-Parameter in einen Parameterset erfolgt mit dem Kommando SP FT%*name*. Näheres siehe Seite **283**. Beim Speichern des Parametersets werden auch evtl. vorhandene FTEP-Parameter berücksichtigt.

host2 Symbolischer Name des fremden Host-Rechners.
Bei der Angabe von *host2* handelt es sich nicht um den Host-Namen laut PDN-Generierung, sondern um den Partnernamen gemäß dem FT-Netzbeschreibungsbuch im aufrufenden System.

user-id2 User-Id im fremden Host-Rechner, in die die Datei zu übertragen ist bzw. aus der die Datei abzuholen ist.
Standard: gleiche Benutzerkennung wie im lokalen System.

FTAC: Alternativ zu einer Remote User-Id kann auch eine FTAC Transfer-Admission für das fremde System angegeben werden. Hierzu ist im Feld "Remote User-ID" die

Konstante "FTAC:" anzugeben, gefolgt von der eigentlichen Transfer-Admission. Bei der Eingabe der Transfer-Admission ist auf die korrekte Groß-/Kleinschreibung entsprechend dem FTAC-Profil zu achten. Von CFS werden eingegebene Kleinbuchstaben nicht in die entsprechenden Großbuchstaben umgewandelt.

acct-nr2 Abrechnungsnummer. Standard: gleiche Abr.Nummer wie im lokalen System.

passw2 Logon-Passwort der Benutzerkennung im fremden System ([C]'...' | X'...').
Standard: kein Passwort.

rem-succ Remote Success-Procedure. Name einer Folgeverarbeitung im Remote-System, die bei erfolgreicher Übertragung ausgeführt wird.

*proc,** Die als *proc* angegebene DO-Prozedur erhält als Parameter den Namen der übertragenen Datei.

/cmd .... Nach erfolgreicher Übertragung wird im Remote-System das BS2000-Kommando ausgeführt. Falls '*' im BS2000-Kommando angegeben wurde, so wird an der entsprechenden Stelle der Name der Remote-Datei substituiert. Es können mehrere BS2000-Kommandos, getrennt durch Semikolons, angegeben werden (*/cmd1* ...;*/cmd2* ...).

'/cmd ...' Nach erfolgreicher Übertragung wird im Remote-System das angegebene Kommando ausgeführt. Das Zeichen '*' wird in dieser Variante nicht durch einen Dateinamen ersetzt.
Standard: keine Remote Success-Procedure.

rem-fail Remote Failure-Procedure. Name einer Folgeverarbeitung im Remote-System die bei nicht erfolgreicher Übertragung ausgeführt wird.

*proc,** Die als *proc* angegebene DO-Prozedur erhält als Parameter den Namen der zu übertragenden Datei.

/cmd .... Bei nicht erfolgreicher Übertragung wird im Remote-System das BS2000-Kommando ausgeführt. Falls '*' im Kommando angegeben wurde, so wird an der entsprechenden Stelle der Name der Remote-Datei substituiert. Es können mehrere BS2000-Kommandos, getrennt durch Semikolons, angegeben werden (*/cmd1* ...;*/cmd2* ...).

'/cmd ...' Bei nicht erfolgreicher Übertragung wird im Remote-System das angegebene Kommando ausgeführt. Das Zeichen '*' wird in dieser Variante nicht durch einen Dateinamen ersetzt.
Standard: keine Remote Failure-Procedure.

rem-par Remote Parameter (Typ des fremden Systems).

mögliche Angaben: <u>*BS2000</u> | *MSP | *ANY | *any

*BS2000 Zielrechner hat das Betriebssystem BS2000.
*MSP Zielrechner hat das Betriebssystem MVS.
*ANY Zielrechner hat ein anderes Betriebssystem (SINIX/VMS/..). Dateinamen werden dem Fremdsystem in <u>Großbuchstaben</u> übergeben.
*any wie *ANY, jedoch werden die BS2000-Dateinamen dem Fremdsystem in <u>Kleinbuchstaben</u> übergeben.

Standard: *BS2000.

direct Transfer-Direction FROM | <u>TO</u> (Richtung des File-Transfer).
Standard: TO.

lst Listing: YES | NO
Soll für jede übertragene Datei ein Protokoll ausgegeben werden ?
Die Angabe YES bewirkt außerdem, daß bei Beendigung des File-Transfer Auftrags das Job-Report System von CFS aktiviert wird, um die von FT erzeugte Ergebnisliste am Bildschirm anzuzeigen.
Standard: NO.

time *hh:mm* Beginn der Übertragung nicht vor der angegebenen Uhrzeit.
Wurde eine Startzeit vor der aktuellen Uhrzeit angegeben, so wird der Auftrag am folgenden Tag zur angegebenen Uhrzeit angestartet.
Standard: Beginn der Übertragung zum frühestmöglichen Zeitpunkt.

" modify Extended Parameters for FT " (letztes Feld der FT-Maske):

Durch die Eingabe Y in diesem Feld wird in die File-Transfer Extended Parameter Maske (FTEP-Maske) verzweigt. In dieser können zusätzliche, jedoch weniger häufig benötigte Parameter für alle folgenden FT-Aufrufe vorbelegt werden. In der FTEP-Maske können z.B. die Parameter "Local Transfer Admission User-Id, Acct-Nr, Password", sowie "Local Success Procedure", "Local Failure Procedure" belegt werden. Die zusätzlichen FT-Parameter können auch vor dem ersten FT-Aufruf durch das Kommando FTEP *param*, siehe Seite 244 vorbelegt werden.
Standard: N (kein Verzweigen in die FTEP-Maske)

Hinweise:

Werden die Parameter bis einschließlich *acct-nr2* in der ONXFT-Anweisung explizit angegeben, so wird die FT-Parameter Maske nicht ausgegeben und für evtl. noch fehlende Parameter werden die Standardwerte angenommen.

Werden PLAM-Elemente in ein anderes BS2000-System übertragen, so werden die Elemente im Zielsystem standardmäßig unter dem gleichen Namen in einer Bibliothek gleichen Namens abgelegt.

Bei der Übertragung von Dateien mit der Variablen Action ONXFT werden Read-Passworte für die zu übertragenden Dateien von CFS nicht nachgefordert.

Die Beendigung von FT-Aufträgen kann mit dem Job-Report System von CFS überwacht werden (FT-Parameter Protocol-Listing: YES). Für weitere Informationen siehe Kommando JRFT/NJRFT auf Seite 412.

Zur Selektion aller mit FT übertragenen Dateien wird auch auf die NSTAT-Option im Feld FILENAME-SELECT der Selektionsmaske hingewiesen.

Für weitere Hinweise zum File-Transfer - insbesondere File-Transfer zwischen BS2000 und SINIX-Systemen - wird auf das Kapitel 12 "File-Transfer mit openFT /RDAC" verwiesen.

Beispiel:

```
ON&FT HOST5,USER5,A001,C'12345',,,,,YES
```
Es werden alle Dateien der Dateienliste zu dem fremden Rechner mit dem Namen HOST5 übertragen. Benutzerkennung, Abrechnungsnummer und Logon-Passwort im fremden System sind: USER5,A001,C'12345'.
Das fremde System ist ein BS2000-Rechner (Standardwert).
Die Übertragung erfolgt in das fremde System (Standard).
Es soll für jede übertragene Datei eine Protokoll-Liste ausgegeben werden.

Guard-Schutz für Dateien / Bibliothekselemente festlegen

```
dd.mm.yy   hh:mm:ss      HOST: ........      TSN: ....  USER-ID: ........

                        modify GUARDS-protection

Read-Guard  (guard-name/*NONE/*SAME) :
Write-Guard (guard-name/*NONE/*SAME) :
Exec-Guard  (guard-name/*NONE/*SAME) :

for Help type ? in any input-field
```

ONX**GUARD** [%*name*|%?[,JRS=*jrfile*]] [R=*guard*|*NONE|*SAME] [,W=*guard*|*NONE|*SAME] [,E=*guard*|*NONE|*SAME]

Die mit X markierten Dateien / PLAM-Bibliothekselemente werden mit einem Guard-Schutz versehen. Falls keine Parameter angegeben sind, werden diese in einer Maske angefordert (siehe oben).

%*name* — Die Parameterangaben für die Variable Action ONXGUARD werden dem angegebenen Parameterset entnommen. Die Speicherung des Parametersets erfolgt mit dem CFS-Kommando SPGUARD %*name*.

%? — Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

JRS=*jrfile* — Name der JRSAVE-Datei, in der der Parameterset gespeichert ist.
Standard: CFS.JRSAVE.*xxxx*. Näheres hierzu siehe Seite 565.

R=*guard* — Der lesende Zugriff auf das Datenobjekt wird mit einem Guard des angegebenen Namens geschützt. Ein Guard kann wie eine Datei mit PVS-ID, User-ID und Name angegeben werden. Der Name ist auf acht Stellen begrenzt.

R=*NONE — Datei: Der lesende Zugriff wird mit einem leeren Guard geschützt und ist daher nicht möglich.
PLAM-Element: Der lesende Zugriff wird weder durch Guard-, noch durch Basic-Access (BACL) Zugriffskontrollen überwacht.

R=*SAME — Die Rechte des lesenden Zugriffs auf das Datenobjekt werden nicht verändert.

W=*guard* — Der schreibende Zugriff auf das Datenobjekt wird mit einem Guard des angegebenen Namens geschützt. Ein Guard kann wie eine Datei mit PVS-ID, User-ID und Name angegeben werden. Der Name ist auf acht Stellen begrenzt.

W=*NONE — Datei: Der schreibende Zugriff wird mit einem leeren Guard geschützt und ist daher nicht möglich.
PLAM-Element: Der schreibende Zugriff wird weder durch Guard-, noch durch Basic-Access (BACL) Zugriffskontrollen überwacht.

W=*SAME — Die Rechte des schreibenden Zugriffs auf das Datenobjekt werden nicht verändert.

E=*guard* — Der ausführende Zugriff auf das Datenobjekt wird mit einem Guard des angegebenen Namens geschützt. Ein Guard kann wie eine Datei mit PVS-ID, User-ID und Name angegeben werden. Der Name ist auf acht Stellen begrenzt.

E=*NONE Datei: Der ausführende Zugriff wird mit einem leeren Guard geschützt und ist daher nicht möglich.
PLAM-Element: Der ausführende Zugriff wird weder durch Guard-, noch durch Basic-Access (BACL) Zugriffskontrollen überwacht.

E=*SAME Die Rechte des ausführenden Zugriffs auf das Datenobjekt werden nicht verändert.

Hinweise:

Für Dateien gilt folgende Vereinbarung: Falls bei allen drei Zugriffsarten (R/W/E) der Wert *NONE eingetragen ist, wird die Guard-Zugriffskontrolle insgesamt ausgeschaltet. Die Dateien sind danach nur noch mit SHARE=YES|NO,ACCESS=READ|WRITE geschützt. Der gleiche Effekt kann auch mit dem Action-Code SGN (Set Guards None) erreicht werden.

Bei dem Schutz von PLAM-Elementen mit der Variablen Action ONXGUARD wird die Existenz der angegebenen Guards nicht geprüft. Wird eine Zugriffsart, z.B. R (Read) mit einem nicht existierenden Guard geschützt, so ist der entsprechende Zugriff (z.B. Read) für alle Benutzer solange gesperrt, bis ein Guard mit dem angegebenen Namen eingerichtet wird.

Beispiel:

```
ON&GUARD R=:C:$TSOS.READG,W=$TSOS.WRITEG,E=$TSOS.EXECG
```

Es werden alle Dateien der Dateienliste mit folgenden Guards geschützt:

Lesender Zugriff:	Guard :C:$TSOS.READG
Modifizierender Zugriff:	Guard $TSOS.WRITEG
Ausführender Zugriff:	Guard $TSOS.EXECG

Dateien mit HSMS aus- bzw. einlagern

ONX**HSMS** [MIGRATE] [, TO=S1|S2] [, COMP=YES|NO] [, INHIBIT=IGNORE|RESPECT]

ONX**HSMS** RECALL [, EXP=YES|NO] [, LOC=BEST|SAME|*cat-id*]

Die Variable Action ONXHSMS gibt dem CFS-Benutzer die Möglichkeit, Dateien mit HSMS auf ein Sicherungsmedium auszulagern (migrate) bzw. von einem Sicherungsmedium zu restaurieren (recall).

MIGRATE Die mit X markierten Dateien (ONX) bzw. alle in der Dateienliste enthaltenen Dateien (ON&) werden mit dem Dienstprogramm HSMS auf ein Sicherungsmedium ausgelagert. In der von CFS generierten Prozedur wird eine HSMS-Anweisung:
//MIGRATE-FILES FROM-STORAGE-LEVEL=*S0-STORAGE-LEVEL(...) erzeugt.
Ab OSD V3.0 ist zu beachten, daß Dateien, die auf verschiedenen SM-Pubsets liegen, in einem HSMS-Lauf nicht zusammen bearbeitet werden können. Dieselbe Einschränkung gilt auch für Dateien, die teilweise auf SF- und SM-Pubsets liegen.

TO=S1|S2 Die Dateien werden auf die angegebene Speicherebene verdrängt.
Dieser Parameter entspricht in der HSMS-Anweisung //MIGRATE-FILES dem Parameter TO-STORAGE=*S1-STORAGE-LEVEL|*S2-STORAGE-LEVEL
Standard: S2

COMP=YES|NO
Bestimmt, ob die Dateien vor dem Schreiben auf den Ausgabedatenträger komprimiert werden oder nicht. Dieser Parameter entspricht in der HSMS-Anweisung //MIGRATE-FILES dem Parameter COMPRESS-FILES=*YES|*NO
Standard: Falls dieser Parameter nicht angegeben wurde, gilt die Standardeinstellung aus der Archivdefinition des Pubsets.

INHIBIT=IGNORE|RESPECT
Dieser Parameter ist nur für den Systemverwalter erlaubt. Es wird festgelegt, ob Migrationssperren, die im Katalogeintrag für die Dateien eingetragen sind, beachtet oder ignoriert werden. Dieser Parameter entspricht in der HSMS-Anweisung //MIGRATE-FILES dem Parameter FILE-INHIBIT=*RESPECTED|*IGNORED
Standard: Falls dieser Parameter unter TSOS nicht angegeben wurde, gilt die Standardeinstellung aus der Archivdefinition des Pubsets.
Unter Nicht-TSOS Kennungen gilt stets der Wert RESPECT.

RECALL Die mit X markierten Dateien (ONX) bzw. alle in der Dateienliste enthaltenen Dateien (ON&) werden mit dem Dienstprogramm HSMS vom Hintergrundspeicher (S1-/S2-Ebene) in die Verarbeitungsebene S0 zurückgeholt. Der Vorteil der Variablen Action ONXHSMS RECALL gegenüber einem impliziten Zurückholen durch Ansprechen (Öffnen) der Dateien besteht darin, daß nicht für jede zu restaurierende Datei ein eigener HSMS-Job erzeugt wird.
In der von CFS generierten Prozedur wird eine HSMS-Anweisung:
//RECALL-MIGRATED-FILES erzeugt.
Ab OSD V3.0 ist zu beachten, daß Dateien, die aus verschiedenen SM-Pubsets migriert wurden, in einem HSMS-Lauf nicht zusammen bearbeitet werden können. Dieselbe Einschränkung gilt auch für Dateien, die teilweise aus SF- und SM-Pubsets migriert wurden.

EXP=YES|NO
Dieser Parameter ist nur für den Systemverwalter erlaubt. Es wird festgelegt, ob der Bandzugriff zu den Zeiten erfolgen soll, die für Expressaufträge vorgesehen wurden.
Dieser Parameter entspricht in der HSMS-Anweisung //RECALL-MIGRATED-FILES dem Parameter OPERATION-CONTROL=*PAR(EXPRESS-REQUEST=*YES|*NO)
Standard: NO

LOC=BEST|SAME|*cat-id*
Dieser Parameter ist erst ab OSD V3.0 erlaubt. Für Dateien, die aus einem SM-Pubset migriert wurden, wird festgelegt, auf welches Volume-Set die Dateien restauriert werden sollen.
BEST Die Dateien werden in Bezug auf die Dateiattribute (z.B. High-Performance, High-Availability) auf das beste, für diese Attribute zur Verfügung stehende Volume-Set zurückgeholt.
SAME Die Dateien werden auf das Volume-Set zurück geholt, auf dem sie vor der Verdrängung gespeichert waren.
cat-id Identifikation des Volume-Sets innerhalb des SM-Pubsets, auf das die Dateien zurückgeholt werden sollen.

Dieser Parameter entspricht in der HSMS-Anweisung //RECALL-MIGRATED-FILES dem Parameter ENVIRONMENT=*SYSTEM-MANAGED(NEW-DATA-LOCATION= *BEST-VOLUME-SET|*SAME|*cat-id*)
Standard: BEST

Hinweise:

Die Variable Action erzeugt eine DO-Prozedur mit den entsprechenden HSMS-Anweisungen. Nach einer Benutzeranfrage kann diese DO-Prozedur über das Job-Report System sofort gestartet und der Ablauf von CFS überwacht werden.

In dem generierten Job werden folgende HSMS-Parameter automatisch eingefügt:
FILE-NAMES=*FROM-FILE(LIST-FILE-NAME=*listfile*)
OPERATION-CONTROL=*PAR(WAIT-FOR-COMPLETION=*YES,REPORT=*FULL, OUTPUT=*protfile*)

Die *listfile* wird von CFS erzeugt und enthält die Namen aller zu bearbeitenden Dateien. Nach dem erfolgreichen HSMS-Lauf wird diese Datei von der Prozedur automatisch gelöscht.

Die *protfile* wird zunächst als temporäre Datei angelegt. Nur wenn der HSMS-Job mit Fehlern beendet wurde, wird die Datei in eine permanente Datei kopiert.

Ab der OSD V3.0 wird bei //MIGRATE-FILES für Dateien, die von einem SM-Pubset gespeichert sind, zusätzlich der Parameter ENVIRONMENT=*SYSTEM-MANAGED (PUBSET-ID=*cat-id*) eingefügt. Als *cat-id* wird die PVS-ID der ersten zu migrierenden Datei eingesetzt. Handelt es sich dagegen bei der ersten zu migrierenden Datei um eine Datei auf einem SF-Pubset, so wird ENVIRONMENT=*SINGLE-FEATURE eingesetzt.

Dateibeschreibungen (INF-Texte) von Datenobjekten kopieren

ONX**ICOPY** *cat1*, *cat2* [, UPD]

cat1, *cat2* Für die mit X markierten Datenobjekte werden die Informationstexte aus dem Informationskatalog *cat1* in den Informationskatalog *cat2* übertragen.

UPD *cat2* wird mit OPEN=INOUT und nicht mit OPEN=OUTPUT (Standardfall) eröffnet.

Hinweis:

Diese Variable Action eignet sich zum Reorganisieren von INF-Katalogen, die möglicherweise Beschreibungen für bereits gelöschte Dateien enthalten. Die aktuellen Dateien, für die Informationstexte vorliegen, können mit der User Option INF Y gezielt ausgewählt werden. Die mit diesen Dateien verknüpften Inf-Texte werden durch ON&ICOPY *cat1*,*cat2* in einen neuen Informationskatalog *cat2* übertragen. Der alte Katalog (z.B. CFS.INF) kann anschließend durch den reorganisierten Katalog überschrieben werden.
Für ausführlichere Informationen zum Thema Dokumentationstexte siehe Seite **421** und folgende (Datei-Informationssystem).

Dateien von Archive-Sicherung auf Platte zurückholen

ONX**IMPORT** ['*str1*'=...] [,TO=PUBLIC|*vsn*|(*vsn*,...)] [,DEVICE=*dev*] [,CONV=<u>NO</u>|STD|CONV] [,REP=YES|<u>NO</u>|ALL|ALLP] [,SPACE=<u>REORG</u>|KEEP] [,PASSWORD=...] [⊔ PARAM ⊔ ....]

Die Variable Action ONXIMPORT kann nur in einer mit DIR *directory* ausgewählten Archive-Directory Liste angegeben werden. Mit IMPORT werden Sicherungsstände von Dateien/Jobvariablen, die mit der Archive-Anweisung EXPORT oder DEXPORT gesichert wurden, unter dem gleichen bzw. mit einem anderen Namen wieder in das System eingespielt.

'*str1*'= ... Namenstransformationsregeln. Näheres siehe Seite **109**.

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

TO=PUBLIC|*vsn*|(*vsn*,...)
Entspricht dem Archive-Operanden TO=... in der Anweisung FILES.
Der Parameter TO= bewirkt, daß beim Restaurieren die Dateien auf dem bzw. den angegebenen privaten Datenträgern bzw. im Public-Speicher eingerichtet werden.

Standard: Bei Fehlen des TO= Operanden, werden die Dateien, die von privaten Datenträgern gesichert wurden, wieder auf die ursprünglichen Datenträger zurückgeschrieben. Dateien, die von gemeinschaftlichen Datenträgern gesichert wurden, werden wieder im Public-Bereich eingerichtet.

DEVICE=*dev* Nur im Zusammenhang mit Parameter TO=*vsn* (s.o.). Hier wird der Gerätetyp des bzw. der bei TO= angegebenen privaten Datenträger angegeben.

CONV=NO|STD|CONV

Entspricht dem Archive-Operanden CONVERSION in der Anweisung [D]IMPORT.
Mit diesem Parameter wird gesteuert, ob Dateien, die vor der Sicherung einen PAM-Schlüssel hatten, konvertiert werden sollen, wenn der Ausgabedatenträger eine NK-Platte ist.

REP=YES|NO|ALL|ALLP

Entspricht dem Archive-Operanden REPLACE=... in der Anweisung [D]IMPORT.
Mit diesem Parameter wird gesteuert, ob vorhandene Dateien/Jobvariablen gleichen Namens beim Rückschreiben ersetzt werden sollen oder nicht. Für weitere Informationen wird auf das ARCHIVE-Handbuch, Anweisung IMPORT verwiesen.

SPACE=REORG|KEEP

Entspricht dem Archive-Operanden SPACE=... in der Anweisung [D]IMPORT.
Mit diesem Parameter wird gesteuert, ob die aus der Sicherung zu ersetzenden Dateien vor dem Zurückschreiben gelöscht werden sollen. Für weitere Informationen wird auf das ARCHIVE-Handbuch, Anweisung IMPORT verwiesen.

PASSWORD=*passw*|(*passw*,...)

Entspricht dem Archive-Operanden PASSWORD=... in der Anweisung [D]IMPORT.
Hier sind ein bzw. mehrere Lesekennworte für gesicherte Dateien anzugeben. Der Systemverwalter braucht die Lesekennworte nicht anzugeben. Für weitere Informationen wird auf das ARCHIVE-Handbuch, Anweisung IMPORT verwiesen.

PARAM ... Entspricht der Archive-Anweisung PARAM in einem IMPORT-Lauf.
Hier können allgemeine Archive-Einstellungen angegeben werden, wie z.B.
OPERATOR=YES|NO,RESTART=YES|NO,WRCHK=YES|NO,SNR=YES|NO,STREAM=YES|NO,UNLOAD=YES|NO Für weitere Informationen zur Bedeutung der einzelnen Parameter wird auf das ARCHIVE-Handbuch, Anweisung PARAM verwiesen.

Der Operand CATID=YES|NO darf in der PARAM-Anweisung nicht angegeben werden, da er von CFS selbst gesetzt wird gemäß der im Archive-Directory vermerkten Einstellung.

Die Anweisung PARAM darf nur als letztes angegeben werden und muß mit einem Blank von den vorhergehenden Parametern getrennt werden.

Dateien / Bibliothekselemente / Jobvariablen in Druckdatei ausgeben

ONX**LIST** [*m*,] *datei* [, E|O] [, TOC] [, HTML] [,L*n*] [, W [*c*]] [, H] [, H4] [, NN] [, RH] [, NH] [, IOH]

m Es werden nur die ersten *m* Sätze der Datenobjekte aufgelistet.

datei Der Inhalt aller mit X gekennzeichneten Datenobjekte wird in eine druckaufbereitete Datei geschrieben.

E Die Ausgabedatei wird mit OPEN=EXTEND eröffnet.
<u>O</u> Die Ausgabedatei wird mit OPEN=OUTPUT eröffnet. (Standard).

TOC Table of Contents. In der Ausgabedatei wird ein Inhaltsverzeichnis aller aufgelisteten Datenobjekte vorangestellt. Das Inhaltsverzeichnis enthält die Namen der Objekte und die Seite, bei der sie in der Liste beginnen.
Standard: kein Inhaltsverzeichnis in der Ausgabeliste.

HTML HTML formatiertes Inhaltsverzeichnis für die Ausgabe der List-Datei durch einen Internet-Browser. Das Inhaltsverzeichnis enthält die Namen der Objekte und einen HTML-Verweis zum Anfang der betreffenden Objekte. Im Browser gelangt man durch einen Mausclick auf einen Namen im Inhaltsverzeichnis direkt zum Anfang des betreffenden Objekts.
Standard: kein Inhaltsverzeichnis in der Ausgabeliste.

L *n* Anzahl der Zeilen pro Druckseite. *n*: zwei- bzw. dreistellige Zahl.
Standard: L=64 Der Systemverwalter kann den Standardwert auf die Gegebenheiten des Rechenzentrums anpassen (CFSMAIN).

W [*c*] Wrap. Datensätze, die länger als *c* Bytes sind, werden in Folgezeilen dargestellt. *c* ist eine zwei bzw. dreistellige Zahl < 256, die die Druckspalte angibt, bei der umgebrochen wird. Falls *c* nicht angegeben, so wird der Standardwert von 132 angenommen. Der Systemverwalter kann den Standardwert auf die Gegebenheiten des Rechenzentrums anpassen (CFSMAIN).

Standard: Wrap-Modus ausgeschaltet.
Jeder Datensatz der aufgelisteten Datei wird in einem Satz der Ausgabedatei dargestellt. Bezüglich der maximalen Satzlänge sind die Einschränkungen durch die verwendete Blocksize zu beachten:

RH bzw. NH-Parameter ist angegeben:
Ein Satz der Ausgabedatei kann bis zu 32749 Byte lang werden. Die BLKSIZE der Ausgabedatei beträgt 16.
Normalmodus, d.h. RH bzw. NH-Parameter ist nicht angegeben:
Es werden bis zu 2026 Datenbytes in den Ausgabesatz übernommen. Die BLKSIZE der Ausgabedatei beträgt 1.

Beim Ausdruck mit SPACE=E wird jeder Satz bei Spalte 132 abgeschnitten.

H Hexadezimale Ausgabe.
Die Ausgabe erfolgt ähnlich wie im Display-Modus von CFS nach Eingabe der Kommandos H und DL (Hexa-Darstellung, Display Long) bzw. wie im BS2000 Dump-Format. In jedem Satz der Ausgabedatei werden 40 Byte links in hexadezimaler Darstellung und rechts in Character-Darstellung ausgegeben.
Standard: Keine hexadezimale Darstellung.

H4 Hexadezimale Ausgabe vierzeilig.
Die Ausgabe erfolgt ähnlich wie im Hexadezimalmodus des EDT. In der Ausgabedatei werden zur Anzeige von 80 Datenbytes vier Druckzeilen benötigt. Zeile 1 enthält ein Zeilenlineal (Spaltenzähler). Zeile 2 enthält die Character-Darstellung der 80 Datenbytes. Zeile 3 und Zeile 4 enthalten das höherwertige/ niederwertige Halbbyte eines jeden Datenbytes. Bei ISAM-Dateien wird die Position des Schlüssels im Zeilenlineal durch die Zeichen KK...KK dargestellt.
Standard: Keine hexadezimale Darstellung.

NN No Numbers. Die Datensätze werden ohne vorangestellte Satznummern aufgelistet.
Standard: Bei SAM-/PAM-Dateien mit vorangestellten Satznummern.

NH No Header. Die Datensätze werden ohne die Überschriftszeile "FILENAME: *xxxx*" sowie ohne Druckaufbereitung (Vorschubsteuerzeichen) aufgelistet. Die No Header-Option der Variablen Action ONXLIST eignet sich dafür, mehrere Dateien in eine se-

quentielle Datei mit dem Datenformat FCBTYPE=SAM,RECFORM=V,BLKSIZE=(STD,1) zusammenzuspielen.
Die NH-Option schließt automatisch die NN-Option (No Numbers) mit ein.

RH Reduced Header. Vor jedem aufgelisteten Datenobjekt wird ein Kopfsatz eingefügt, der den Datei- bzw. Bibliotheks- und Elementnamen enthält. Die eigentlichen Sätze des Datenobjekts werden wie bei NH (No Header) ohne Druckaufbereitung (Vorschubsteuerzeichen) aufgelistet.

IOH Insert only Header. Die aufzulistenden Dateien sind druckaufbereitet. Die aufzulistenden Dateien werden von CFS unverändert mit den bestehenden Vorschubsteuerzeichen übernommen. CFS fügt vor jeder Datei eine zusätzliche Seite ein, die den Header enthält.

Hinweise:

Bei TOC werden die Parameter NH / RH nicht ausgewertet.

Bei H bzw. H4 ist der Parameter W (Wrap) standardmäßig eingeschaltet.

PAM-Dateien können mit ONXLIST nur im H- bzw. H4-Modus aufgelistet werden.

Bei der Auflistung des Inhalts von Jobvariablen sollte der Parameter RH (Reduced Header) angegeben werden: ONXLIST *ausdat*,RH.

Katalogeinträge von Dateien/Jobvariablen in Datei sichern

ONX**LISTCAT** *ausdat* [, VERSION=0 | 2 | 3] [, E]

ausdat Diese Variable Action schreibt für die mit X gekennzeichneten Dateien/Jobvariablen den vollen Katalogeintrag als einen Satz variabler Länge in die Datei *ausdat*.

VERSION=0 Dieser Parameter dient vor allen für Diagnosezwecke und bewirkt, daß der vollständige Katalogeintrag im alten, BS2000 V10-kompatiblen Format in die Listcat-Datei geschrieben wird. Eine mit VERSION=0 erstellte Listcat-Datei kann mit dem Kommando RECAT verarbeitet werden. Die Wiederherstellung der Schutzattribute Guards ist damit jedoch nicht möglich.

VERSION=2 Dieser Parameter ist ab OSD V1 anwendbar und dient vor allen für Diagnosezwecke. Die Angabe bewirkt, daß der Katalogeintrag im Format des neuen FSTAT-Layouts auf der Basis von OSD V3 ausgegeben wird. Eine mit VERSION=2 erstellte Listcat-Datei kann nicht mit dem Kommando RECAT verarbeitet werden.

VERSION=3 Dieser Parameter steht nur dem Systemverwalter unter der Kennung TSOS zur Verfügung. Es wird der physikalische Katalogeintrag in die Ausgabedatei geschrieben. Das Layout wird durch die DSECT DCACATE MF=D,CETYPE=*FILE | *JV beschrieben.
Eine mit VERSION=3 erstellte Listcat-Datei kann nicht mit dem Kommando RECAT verarbeitet werden.

Bei einer geplanten späteren Verarbeitung der Listcat-Datei durch das CFS-Kommando RECAT (siehe Seite **Fehler! Es wurde kein Textmarkenname vergeben.**), sollte der Parameter VERSION= nicht angegeben werden.

E Extend. Dieser wahlfreie Zusatz bewirkt, daß die Ausgabedatei mit OPEN=EXTEND eröffnet wird.

Hinweise:

Bei der Aufzeichnung der Katalogeinträge von Jobvariablen mit der Variablen Action LISTCAT werden neben den Attributen auch die Daten der Jobvariablen festgehalten. Mit dem Zusatzparameter CJ (Create Jobvariables) im RECAT-Kommando ist es möglich, Jobvariablen mit den originalen Daten wieder zu erzeugen. Es können damit z.B. **Jobvariablen auf anderen Host-Rechnern eingerichtet** werden: Listcat-Datei per File-Transfer übertragen; auf dem fremden Host: RECAT *datei*,CJ.

Das Kommando **RECAT** kann die mit LISTCAT erzeugte Katalogauszugsdatei verarbeiten. RECAT prüft die in der LISTCAT-Datei festgehaltenen Dateiattribute mit den aktuellen Dateiattributen. Bei Ungleichheit werden geeignete CAT-Kommandos abgesetzt, um die früheren Dateiattribute wieder herzustellen. Für eine genaue Beschreibung des RECAT-Kommandos siehe Seite **Fehler! Es wurde kein Textmarkenname vergeben.**.

Dateien auf andere Platte/auf ein anderes Public-Volume-Set (PVS) übertragen

ONX**MOVE** *:x:* [,*smspar*] [,*vsn/device*] | *vsn/device* | [PUBLIC | DEFCAT] [,*smspar*] [,A] [,ED] [,KD]

:x: Die vom Benutzer mit dem Action-Code X bezeichneten Dateien werden unter gleichem Namen auf den angegebenen Public-Volume-Set übertragen.

Die Variable Action ONXMOVE verarbeitet keine Jobvariablen. Zum Übertragen von Jobvariablen auf einen anderen Pubset muß die Variable Action ONXCOPY *:x:* verwendet werden. Durch Eintragen des Action-Codes XE können die JVs auf dem Ursprungspubset nach erfolgreichem Kopieren gelöscht werden.

smspar Falls die Datei auf einen SM-Pubset (System Managed Pubset) übertragen wird, so kann mit diesem Parameter in Abhängigkeit von den Vorgaben des Systemverwalters der Ablageort (Volume-Set) der Datei genauer bestimmt werden.
Falls vor *smspar* der Parameter *:x:* nicht angegeben wurde, so bezieht sich *smspar* auf den Default-Pubset der Benutzerkennung.

AVAILABILITY=HIGH
Die Datei soll innerhalb eines SM-Pubsets auf einem Volume-Set abgelegt werden, der erhöhte Ausfallsicherheit besitzt (z.B. durch DRV-gespiegelte Platten).

STORAGE-CL=*class*
Name einer vom Systemverwalter eingerichteten Storage Klasse. Über die Storage Klasse wird das Volume-Set der Datei und damit die Speichereigenschaften wie z.B. Performance Attribute, Availability festgelegt.

VOLUME-SET=*vset* [,S0-MIGRATION=*ALLOWED|*FORBIDDEN]
Bestimmt das Volume-Set, auf dem die Datei innerhalb des SM-Pubsets eingerichtet werden soll. Der Systemverwalter kann mit *CONTROL bzw. mit der Angabe einer ein- bis vierstelligen Cat-Id *vset* (ohne ':') einen Volume-Set explizit angeben.
Der nichtprivilegierte Benutzer kann einen Volume-Set nur angeben, wenn ihm der Systemverwalter das Recht zur physikalischen Allokierung gegeben hat.
VOLUME-SET=*vset*,S0-MIGRATION=*ALLOWED
Standardmäßig wird bei MOVE auf ein bestimmtes Volume-Set eines PVS das Attribut S0-MIGRATION auf FORBIDDEN gesetzt. Durch den Parameter S0-MIGRATION

kann die Eigenschaft nun auf ALLOWED gesetzt werden, wie dies bei der Datei vor der MOVE-Operation in der Regel der Fall ist.

WORK-FILE=YES|NO
Gibt an, ob die Datei eine Arbeitsdatei ist, die von der Systemverwaltung zu einem bestimmten Zeitpunkt gelöscht werden darf. Work-Dateien werden auf einem Volume-Set abgelegt, für das der Systemverwalter die Nutzungsart WORK vorgegeben hat.

vsn/device Die mit dem Action-Code X angekreuzten Dateien werden unter gleichem Namen auf der angegebenen Privatplatte eingerichtet und auf dem ursprünglichen Datenträger gelöscht.

PUBLIC Die mit dem Action-Code X angekreuzten Dateien werden unter gleichem Namen im Public-Bereich eingerichtet und auf dem ursprünglichen Datenträger gelöscht.

DEFCAT Die mit dem Action-Code X angekreuzten Dateien werden unter gleichem Namen im Public-Bereich des Default-Pubsets der entsprechenden Benutzerkennung eingerichtet und auf dem ursprünglichen Datenträger gelöscht.

A All. Die Move-Operation wird auch für leere Dateien durchgeführt.

ED Erase Destroy. Auf dem Ursprungsdatenträger werden die Dateien beim Löschen mit binären Nullen überschrieben.

KD AB OSD V8.0: Das Change-Date der Originaldatei wird auf die neue Datei übertragen.

Hinweise:

Die Variable Action kann auf Dateigenerationen nur in der Form ONXMOVE *:x:* angewendet werden. Um eine Dateigeneration auf ein anderes PVS zu übertragen, sollte diese mit dem Parameter VOLUME : FGG ausgewählt werden. ONXMOVE *:x:* löscht die Dateigenerationen auf dem Ursprungspubset nicht (gleiche Funktion wie ONXCOPY "='*:x:*').

Beim Moven von Dateien mit BLKCTRL=PAMKEY auf ein Nonkey PVS wird von der Variablen Action intern das Programm PAMCONV aufgerufen.

Beispiele:

`ONXMOVE PVT001/D348E`
Die durch X markierten Dateien werden auf der Privatplatte PVT001 eingerichtet und auf den ursprünglichen Datenträgern gelöscht.

`ONXMOVE :G:`
Die durch X markierten Dateien werden auf dem PVS :G: eingerichtet und auf dem ursprünglichen PVS gelöscht.

Geöffnete Dateien als sicherungswürdig kennzeichnen

ONX**OPNBACK** [= <u>Y</u> | N]

Eine Datei, die im Schreib- oder Änderungsmodus eröffnet ist (Open=Output/Update /Extend/Inout/Outin), kann im Normalfall von ARCHIVE nicht gesichert werden. Mit dieser Variablen Action wird in einem Bit im TSOSCAT vermerkt, daß die Datei auch im Zustand der Änderung gesichert werden soll. Voraussetzung hierfür ist BS2000 ab OSD V1.0 und daß PARAM OLS=YES angegeben wurde. Der Benutzer ist in diesem Falle selbst für die Konsistenz der gesicherten Daten verantwortlich.

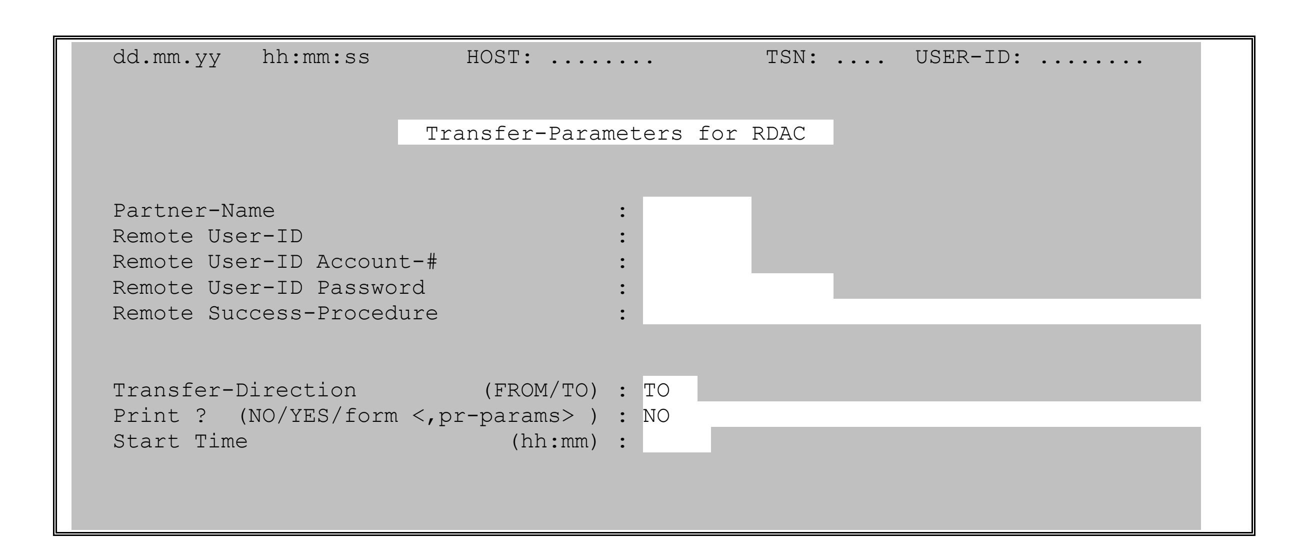
Zum Zeitpunkt der Ausführung der Variablen Action OPNBACK darf die Datei nicht gesperrt sein. Gegebenenfalls ist zuvor die Variable Action ONXVERIFY anzuwenden.

Die Möglichkeit, Dateien die sich im Zustand der Änderung befinden zu sichern, wurde primär für UDS-Datenbanken geschaffen. Eine sinnvolle Anwendung ergibt sich aber z.B. auch für PLAM-Bibliotheken, in die z.B. SYSOUT-Dateien geschrieben werden. Im ungünstigsten Fall befindet nach dem Restore einer solchen Bibliothek ein Element (SYSOUT-Protokoll) in einem nicht konsistenten Zustand. Ohne diese Option könnte die Bibliothek überhaupt nicht oder nur sehr selten gesichert werden.

Y
: Der OPNBACK-Indikator soll für die angekreuzten Dateien im Systemkatalog permanent gesetzt werden.

N
: Der OPNBACK-Indikator soll für die angekreuzten Dateien zurückgesetzt werden.

Dateien mit RDAC in andere Host-Rechner / PC übertragen
Dateien / Bibliothekselemente an Stationsdrucker ausdrucken



```
dd.mm.yy   hh:mm:ss      HOST: ........        TSN: ....  USER-ID: ........

                        Transfer-Parameters for RDAC

Partner-Name                                :
Remote User-ID                              :
Remote User-ID Account-#                    :
Remote User-ID Password                     :
Remote Success-Procedure                    :

Transfer-Direction          (FROM/TO)       : TO
Print ?  (NO/YES/form <,pr-params> )        : NO
Start Time                    (hh:mm)       :
```

ONX**RDAC** ['*str1*'=...] [*host2*] [*,user-id2*] [*,acct-nr2*] [*,passw2*] [*,rem-succ*] [*,direct*] [*,print*] [*,time*]

Falls keine Operanden angegeben sind, werden diese in einer FT-Parameter Maske angefordert (siehe oben).

'*str1*'= ...
: Namenstransformationsregeln. Näheres siehe Seite **Fehler! Es wurde kein Textmarkenname vergeben.**.

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

Enthält der Name der Ursprungsdatei die Zeichenfolge '*str1*' nicht, so erfolgt kein File-Transfer. Fehlt die Angabe '*str1*'=..., so erhalten die Dateien im Zielrechner die gleichen Namen wie im Ursprungssystem.

host2
: Symbolischer Name des fremden Host-Rechners.
Bei der Angabe von *host2* handelt es sich nicht um den Host-Namen laut PDN-Generierung, sondern um den Partnernamen laut RDAC-Generierung im aufrufenden System.

user-id2 User-Id im fremden Host-Rechner, in die die Datei zu übertragen ist bzw. aus der die Datei abzuholen ist.
Standard: gleiche Benutzerkennung wie im eigenen Host.

acct-nr2 Abrechnungsnummer.
Standard: gleiche Abrechnungsnummer wie im eigenen Host.

passw2 Logon-Passwort der Benutzerkennung im fremden System ([C]'...' | X'...').
Standard: kein Passwort.

rem-succ Remote Success-Procedure. Name einer Folgeverarbeitung im Remote-System, die bei erfolgreicher Übertragung ausgeführt wird.

*proc,** Die als *proc* angegebene DO-Prozedur erhält als Parameter den Namen der übertragenen Datei.

/cmd .... Nach erfolgreicher Übertragung wird im Remote-System das angegebene BS2000-Kommando ausgeführt. Falls '*' im Kommando angegeben wurde, so wird an der entsprechenden Stelle der Name der Remote-Datei substituiert.

'/cmd ...' Nach erfolgreicher Übertragung wird im Remote-System das angegebene Kommando ausgeführt. Das Zeichen '*' wird in dieser Variante nicht durch einen Dateinamen ersetzt.
Standard: keine Remote Success-Procedure.

direct Transfer-Direction FROM | <u>TO</u> (Richtung des File-Transfer).
Standard: Transfer-Direction : TO.

print Soll die Datei zu einem Remote-Drucker übertragen werden ?
<u>NO</u> | YES | *form* [, *pr-params*]

Die Angabe eines Wertes ungleich YES/NO wird als Formularbezeichnung interpretiert und bewirkt einen Ausdruck mit dem angegebenen Form-Parameter.

Im Anschluß an die Werte YES | *form* können wahlweise ein oder mehrere Print-Parameter angegeben werden:

pr-params SPACE= | COPIES= | HEADER= | LINES=nn
Standard: Print ? : NO.

time *hh:mm* Beginn der Übertragung nicht vor der angegebenen Uhrzeit.
Standard: Beginn der Übertragung zum frühestmöglichen Zeitpunkt.

Hinweise:

Werden die Parameter bis einschließlich *acct-nr2* in der ONXRDAC-Anweisung angegeben, so wird die Parameter-Maske des File-Transfer nicht angezeigt. Für fehlende Parameter werden die Standardwerte eingesetzt.

Für weitere Hinweise zum File-Transfer mit RDAC - insbesondere File-Transfer zwischen BS2000 und MS-DOS-Systemen wird auf das Kapitel 12 "File-Transfer mit openFT/RDAC" verwiesen.

Dateien / Bibliothekselemente / Jobvariablen umbenennen

ONX**REN** '*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

'*str1*'='*str2*'
Die mit X gekennzeichneten Datenobjekte werden umbenannt. Der neue Name wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und durch '*str2*' ersetzt wird. Definitionsgemäß wird der String '␣' am Ende und der Nullstring '' am Anfang eines jeden Namens gefunden.

'*str1*'=DP[*n*][I]
Delete Prefix: Der neue Name des Datenobjekts wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der links davorstehende Namensteil bzw. die *n* links davorstehenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht.

'*str1*'=DS[*n*][I]
Delete Suffix: Der neue Name des Datenobjekts wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der nachfolgende Namensteil bzw. die *n* nachfolgenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht. Näheres siehe Seite **Fehler! Es wurde kein Textmarkenname vergeben.**.

Enthält der Name die Zeichenfolge '*str1*' nicht, so wird die Variable Action für dieses Datenobjekt nicht ausgeführt.

Hinweise:

Die Benutzerkennung kann mit der Variablen Action REN nicht verändert werden, d.h. es ist kein Umbenennen zwischen verschiedenen User-IDs möglich.

Beim Umbenennen von Elementen einer PLAM-Bibliothek sind folgende Sonderfälle zu beachten:
ONXREN '␣'='*suffix*' ein Blank in String-1
Bei Elementen mit Versionen wird die angegebene Suffix nach dem Elementnamen und vor der Versionsbezeichnung eingefügt.
ONXREN '␣'='/*suffix*' ein Blank in String-1
Bei Elementen mit Versionsbezeichnung wird diese entfernt und durch die angegebene Version /suffix ersetzt.
Bei Elementen ohne Version wird die angegebene Version /suffix an den Elementnamen angehängt.
ONXREN '␣␣'='*suffix*' zwei Blanks in String-1
Bei Elementen mit Versionen wird die angegebene Suffix an die bestehende Versionsbezeichnung angehängt.

Beim Umbenennen von Elementen einer PLAM-Bibliothek kann durch '/'='/*prefix*.' ein Prefix im Namen vorangestellt werden.
Beispiel: ONXREN'/'='/NEU.' (S/XYZ/001 --> S/NEU.XYZ/001)

Beispiele:

`ONXREN'KLM'='S'`
Beim ersten Auftreten des Strings 'KLM' wird dieser in 'S' umgewandelt.
z.B. AB.D.KLMDATA --> AB.D.SDATA

`ONXREN''='ABC.'`
Dem Namen wird das Prefix 'ABC.' vorangestellt.

`ONXREN' '='.X'`
Dem Namen wird das Suffix '.X' nachgestellt.

`ONXREN' '=DP7`
Die jeweils letzten sieben Stellen im Dateinamen werden entfernt.
z.B. ABC.S.TEST.950510 --> ABC.S.TEST

`ONXREN'   /'='001/'`
Mit dieser Variablen Action wird bei Elementen einer LMS-ISAM Bibliothek die Versionsbezeichnung Blank (.../ ⊔ ⊔ ⊔/...) durch 001 ersetzt (.../001/...).

`ONXREN'/#'=''`
Mit dieser Variablen Action wird eine beliebige Versionsbezeichnung in PLAM-Elementen entfernt.

`ONXREN'/#'='/V100'`
Bei PLAM-Elementen, die bereits eine beliebige Versionsbezeichnung besitzen, wird diese durch /V100 ersetzt.

`ONXREN' '='/V100'`
Bei PLAM-Elementen mit Version wird diese durch /V100 ersetzt.
Bei PLAM-Elementen ohne Version wird die Version /V100 angehängt.

`ONXREN'  '='V100'`
Bei PLAM-Elementen mit Version wird V100 an die bestehende Versionsbezeichnung angehängt. Bei PLAM-Elementen ohne Version wird V100 an den Elementnamen angehängt.

Bibliotheken / ZIP-Archive / Dateien reorganisieren

ONX**REORG** ['*str1*'=...] [, *par*]

Die angekreuzten Datenobjekte werden reorganisiert, indem alle Bibliothekselemente /Datensätze in eine neue Bibliothek/Datei übertragen werden.
Durch die Variable Action REORG werden die durch häufiges Löschen und Neuaufnehmen von Elementen entstandenen "Löcher" aus der Bibliothek entfernt. Die Bibliothek wird "gepackt" und beansprucht am Ende weniger Speicherplatz als vor der Reorganisation. Durch die Defragmentierungsmöglichkeit (Parameter DF) kann auch die Anzahl der von der Datei beanspruchten Extents verringert werden.

'*str1*'= ... Namenstransformationsregeln

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

'*str1*'='*str2*'
Der Name des reorganisierten Datenobjekts wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und durch '*str2*' ersetzt wird. Definitionsgemäß wird der String '⊔' am Ende und der Nullstring '' am Anfang eines jeden Namens gefunden.

'*str1*'=DP[*n*][I]
Delete Prefix: Der Name des reorganisierten Datenobjekts wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der links davorstehende Namensteil bzw. die *n* links davorstehenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht.

'*str1*'=DS[*n*][I]
Delete Suffix: Der Name des reorganisierten Datenobjekts wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der nachfolgende Namensteil bzw. die *n* nachfolgenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht. Näheres siehe Seite **Fehler! Es wurde kein Textmarkenname vergeben.**.

Enthält der ursprüngliche Name die Zeichenfolge '*str1*' nicht, so wird die Variable Action für dieses Datenobjekt nicht ausgeführt.

Fehlt der Operand '*str1*'=..., so wird die erzeugte Zwischenbibliothek (Hilfsdatei) auf das ursprüngliche Datenobjekt zurückkopiert. Nicht benötigter Speicherplatz wird freigegeben (/FILE..,SPACE=-9999). Die Hilfsdatei wird anschließend gelöscht.

par [*vsn*/*device*] | [*:x:*] [,*smspar*] [, PAD=*nn*] [, NR] [, DF[O]] [, SPACE=(*xxxx*,*yyy*)] [, MODE=CFS|LMS]

vsn/device Die reorganisierte Bibliothek/die zur Reorganisation benötigte Hilfsdatei wird auf der angegebenen Privatplatte eingerichtet.

Hinweis:

Falls sich die Hilfsdatei über mehrere Privatplatten erstrecken soll oder falls sie auf einem anderen speziellen Speichermedium eingerichtet werden soll, so kann dies auf die folgende Weise erreicht werden:
Der Benutzer legt auf den gewünschten Datenträgern eine Datei mit beliebigem Namen (Ausnahme: CFS.REORGFILE) und in ausreichender Größe an. Diese Datei wird CFS durch das Kommando /FILE *datei*,LINK=CFSREORG bekannt gemacht. In der Folge werden alle Reorganisationen mit dieser Datei als Hilfsdatei durchgeführt. Die Hilfsdatei wird nach Abschluß der Reorganisation nicht gelöscht, wie dies normalerweise der Fall ist. Es wird lediglich ein Kommando /ERASE *datei*,DATA zum Löschen der Dateninhalte ausgeführt. Der reservierte Speicherplatz bleibt hierdurch bestehen.

:x: Die reorganisierte Bibliothek/die zur Reorganisation benötigte Hilfsdatei wird auf dem angegebenen Public-Volume-Set eingerichtet.

smspar Falls die reorganisierte Bibliothek/die zur Reorganisation benötigte Hilfsdatei auf einem SM-Pubset (System Managed Pubset) angelegt wird, so kann mit diesem Parameter in Abhängigkeit von den Vorgaben des Systemverwalters der Ablageort (Volume-Set) der Datei genauer bestimmt werden.

AVAILABILITY=HIGH
Die Datei soll innerhalb eines SM-Pubsets auf einem Volume-Set abgelegt werden, der erhöhte Ausfallsicherheit besitzt (z.B. durch DRV-gespiegelte Platten).

STORAGE-CL=*class*
Name einer vom Systemverwalter eingerichteten Storage Klasse. Über die Storage Klasse wird das Volume-Set und damit die Speichereigenschaften der Datei (z.B. Performance Attribute, Availability) festgelegt.

VOLUME-SET=*vset*
Bestimmt das Volume-Set, auf dem die Datei innerhalb des SM-Pubsets eingerichtet werden soll. Der Systemverwalter kann mit *CONTROL

bzw. mit der Angabe einer ein- bis vierstelligen Cat-Id *vset* (ohne ':') einen Volume-Set explizit angeben.
Der nichtprivilegierte Benutzer kann einen Volume-Set nur dann angeben, wenn ihm der Systemverwalter das Recht der physikalischen Allokierung erteilt hat.

WORK-FILE=YES|NO
Gibt an, ob die Datei eine Arbeitsdatei ist, die von der Systemverwaltung zu einem bestimmten Zeitpunkt gelöscht werden darf. Work-Dateien werden auf einem Volume-Set abgelegt, für das der Systemverwalter die Nutzungsart WORK vorgegeben hat.

PAD=*nn* Optionaler PAD-Faktor für die reorganisierte Datei. Diese Angabe wird nur bei der Reorganisation von ISAM-Dateien berücksichtigt. PAD=*nn* kann auch nach *:x:* bzw. *vsn/device* angegeben werden (getrennt durch Komma).
Standard: PAD=0

NR No Release of free Space. Zugewiesener, jedoch nicht beschriebener Speicherplatz wird nach Abschluß der Reorganisation nicht freigegeben. Dieser Speicherplatz steht der Datei/Bibliothek für spätere Erweiterungen zur Verfügung.
Standard: Der nicht benötigte Speicherplatz wird freigegeben.

DF[O] Defragment [Only]. Vor dem Zurückkopieren der reorganisierten Hilfsdatei wird der Speicherplatz der Originaldatei mit /ERASE ..., SPACE freigegeben und in der benötigten Größe neu angefordert. Es läßt sich damit in der Regel die Anzahl der Extents der Originaldatei verringern. Die Defragmentierung ist besonders effizient, wenn die Hilfsdatei auf einem anderen Pubset oder auf einer Privatplatte angelegt wurde (ONXREORG :*x*:,DF).
DFO: Es wird nur die Defragmentierung durchgeführt. "Löcher" in Bibliotheken/ISAM-Dateien werden hierbei nicht freigegeben.
Standard: Defragmentierung ausgeschaltet bei Bibliotheken und ISAM-Dateien. Defragmentierung eingeschaltet bei allen übrigen Dateien.

SPACE= Nach erfolgter Reorganisation wird ein FILE-Kommando mit dem angegebenen SPACE-Parameter ausgeführt. Es kann damit überschüssiger Speicherplatz freigegeben und gleichzeitig die Sekundärzuweisung für die Datei modifiziert werden (z.B. SPACE=(-9999,30)).
Standard: Der nicht benötigte Speicherplatz wird freigegeben. Die Sekundärzuweisung wird nicht verändert.

MODE=CFS|LMS
Dieser Parameter legt die angewendete Methode der Reorganisation von PLAM-Bibliotheken fest.

MODE=CFS CFS-eigene, gründliche Reorganisation. Hierbei werden alle Elemente in eine neue Bibliothek übertragen. An Ende wird die neu erzeugte Hilfsbibliothek auf die zu reorganisierende Bibliothek zurückkopiert und danach gelöscht. Diese Methode bringt optimale Reorganisationsergebnisse, nimmt aber besonders bei großen Bibliotheken deutlich mehr Zeit in Anspruch als die LMS-Methode (s.u.).

MODE=LMS LMS bietet ab Version 3.0 eine eigenständige Reorganisationsmethode an. Diese arbeitet inplace, d.h. ohne die einzelnen Elemente zu übertragen und am Ende die Bibliothek zu kopieren. Mit dieser Methode wird eine Bibliothek in kürzestmöglicher Zeit reorganisiert. Unter Umständen werden weniger Seiten freigegeben als bei der CFS-Reorganisation. MODE=LMS wird ignoriert, falls die PLAM-Bibliothek mit Hilfe eines Umbenennungsmusters in eine zweite Bibliothek reorganisiert wird.

Standard: MODE=CFS.
Über CFSMAIN kann der Standard aber auch auf MODE=LMS eingestellt werden.

Hinweise:

ONXREORG... reorganisiert auch beliebige SAM-/ISAM-Dateien. Alle Datensätze werden dazu in eine neu angelegte Zwischendatei übertragen.

Die Variante '*str1*'=... der Variablen Action REORG hat gegenüber der einfachen Variante (Zurückkopieren auf die Originalbibliothek) den Vorteil, daß die neu erzeugte Bibliothek auch die Anzahl der Extents minimiert wird.

Beim Reorganisieren von PLAM-Bibliotheken wird das System-Datum aller Elemente auf den aktuellen Tag gesetzt. Das User-Datum der Elemente wird beim Reorganisieren nicht verändert.

Beispiele:

```
ONXREORG ' '='.REO'
```
Die mit X angekreuzten Bibliotheken werden reorganisiert in neue Bibliotheken, deren Namen am Ende die Zeichenfolge '.REO' enthält.

z.B. CFS.S.LMSLIB --> CFS.S.LMSLIB.REO.

```
ONXREORG''='REORG.'
```
Die mit X markierten Bibliotheken werden reorganisiert in neue Bibliotheken, deren Namen die Zeichenfolge 'REORG.' vorangestellt wird.

z.B. CFS.S.LMSLIB --> REORG.CFS.S.LMSLIB.

```
ONXREORG PVT001/D3475,DF
```
Die markierten Bibliotheken/Dateien werden reorganisiert und defragmentiert. Die für die Reorganisation notwendigen Hilfsdateien werden auf der Privatplatte PVT001 angelegt. Am Ende der Reorganisation wird der Speicherplatz der Originaldatei freigegeben und danach in der benötigten Größe wieder neu angefordert. Die Hilfsdatei auf der Privatplatte wird dann auf die Originaldatei zurückkopiert und auf der Privatplatte gelöscht.

Dateien von Archive-Sicherung auf Platte zurückholen

ONX**RESTORE** ['*str1*'=...,] [STATE] [,TO=PUBLIC|*vsn*|(*vsn*, ...)] [,DEVICE=*dev*] [,ENV-ATT=<u>FROM-TARGET</u>|FROM-ORIGIN] [,CONV=<u>NO</u>|STD|CONV] [,REP=YES|<u>NO</u>|ALL|ALLP] [,SPACE=<u>REORG</u>|KEEP] [,PASSWORD= ...] [,DIR=NONE] [⊔ PARAM ⊔ ....]

Die Variable Action ONXRESTORE kann nur in einer mit DIR *directory* ausgewählten Archive-Directory Liste angegeben werden. Mit RESTORE werden Sicherungsstände von Dateien/Jobvariablen, die mit der Archive-Anweisungen SAVE/DSAVE gesichert wurden, unter dem gleichen bzw. mit einem anderen Namen wieder in das System eingespielt.

'*str1*'= ... Namenstransformationsregeln. Näheres siehe Seite **Fehler! Es wurde kein Textmarkenname vergeben.**.

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

STATE Entspricht dem Archive-Operanden FROM=...,STATE in der Anweisung RESTORE/DRESTORE. Der Parameter STATE hat eine Bedeutung nur bei Sicherungseinträgen

vom Typ PART (Teilsicherung). Er bewirkt, daß beim Restaurieren einer Datei zunächst der letzte Vollsicherungsstand der Datei und anschließend die in der Teilsicherung geänderten Seiten eingespielt werden.

TO=PUBLIC|*vsn*|(*vsn*,...)
Entspricht dem Archive-Operanden TO=... in der Anweisung FILES.
Der Parameter TO= bewirkt, daß beim Restaurieren die Dateien auf dem bzw. den angegebenen privaten Datenträgern bzw. im Public-Speicher eingerichtet werden.

Standard: Bei Fehlen des TO= Operanden, werden die Dateien, die von privaten Datenträgern gesichert wurden, wieder auf die ursprünglichen Datenträger zurückgeschrieben. Dateien, die von gemeinschaftlichen Datenträgern gesichert wurden, werden wieder im Public-Bereich eingerichtet.

DEVICE=*dev* Nur im Zusammenhang mit Parameter TO=*vsn* (s.o.). Hier wird der Gerätetyp des bzw. der bei TO= angegebenen privaten Datenträger angegeben.

ENV-ATT=FROM-TARGET|FROM-ORIGIN
Entspricht dem Archive-Operanden ENVIRONMENT-ATTRIBUTES in der Anweisung [D]RESTORE.
Mit diesem Parameter wird gesteuert, wie RESTORE mit Dateien verfährt, die bereits existieren und mit ACL, BACL oder GUARDS geschützt sind.

CONV=NO|STD|CONV
Entspricht dem Archive-Operanden CONVERSION in der Anweisung [D]RESTORE.
Mit diesem Parameter wird gesteuert, ob Dateien, die vor der Sicherung einen PAM-Schlüssel hatten, konvertiert werden sollen, wenn der Ausgabedatenträger eine NK-Platte ist.

REP=YES|NO|ALL|ALLP
Entspricht dem Archive-Operanden REPLACE=... in der Anweisung [D]RESTORE.
Mit diesem Parameter wird gesteuert, ob vorhandene Dateien/Jobvariablen gleichen Namens beim Rückschreiben ersetzt werden sollen oder nicht. Für weitere Informationen wird auf das ARCHIVE-Handbuch, Anweisung RESTORE verwiesen.

SPACE=REORG|KEEP
Entspricht dem Archive-Operanden SPACE=... in der Anweisung [D]RESTORE.
Mit diesem Parameter wird gesteuert, ob die aus der Sicherung zu ersetzenden Dateien vor dem Zurückschreiben gelöscht werden sollen. Für weitere Informationen wird auf das ARCHIVE-Handbuch, Anweisung RESTORE verwiesen.

PASSWORD=*passw*|(*passw*,...)
Entspricht dem Archive-Operanden PASSWORD=... in der Anweisung [D]RESTORE.
Hier sind ein bzw. mehrere Lesekennworte für gesicherte Dateien anzugeben. Der Systemverwalter braucht die Lesekennworte nicht anzugeben. Für weitere Informationen wird auf das ARCHIVE-Handbuch, Anweisung RESTORE verwiesen.

DIR=NONE Dieser Parameter bewirkt, daß beim Restore mit ARCHIVE keine Directory-Datei benötigt wird. Dies bedeutet, daß die erzeugte DO-Prozedur in der ARCHIVE Restore-Anweisung die VSN's der Bänder enthält, die zum Rücksichern der Dateien benötigt werden sowie den entsprechenden Device-Typ. Als Hinweis an den Operator zum Bereitstellen der benötigten Bänder werden per TYPE entsprechende Meldungen an CONSOLE ausgegeben.

Ein Anwendungsfall für den Parameter DIR=NONE besteht darin, an einem Host eine ARCHIVE Restore-Prozedur zum Zurücksichern bestimmter Dateien zu erzeugen. Diese Prozedur kann mit File-Transfer auf einen anderen Host übertragen und dort zur Ausführung gebracht werden. Die gewünschten Dateien werden auf dem fremden

Host ohne Directory-Datei zurückgesichert. Vorausgesetzt wird lediglich der Zugriff auf die benötigten Archive-Bänder.

PARAM ... Entspricht der Archive-Anweisung PARAM in einem RESTORE-Lauf.
Hier können allgemeine Archive-Einstellungen angegeben werden, wie z.B. OPERATOR=YES|NO,RESTART=YES|NO,WRCHK=YES|NO,SNR=YES|NO,STREAM=YES|NO,UNLOAD=YES|NO. Für weitere Informationen zur Bedeutung der einzelnen Parameter wird auf das ARCHIVE-Handbuch, Anweisung PARAM verwiesen.

Der Operand CATID=YES|NO darf in der PARAM-Anweisung nicht angegeben werden, da er von CFS selbst gesetzt wird gemäß der im Archive-Directory vermerkten Einstellung.

Die Anweisung PARAM darf nur als letztes angegeben werden und muß mit einem Blank von den vorhergehenden Parametern getrennt werden.

Hinweise:

Mit ONXRESTORE können nur Dateien/Jobvariablen aus einer Archive-Directory Liste von CFS auf Platte zurückgeholt werden, die mit der Archive-Anweisung SAVE bzw. DSAVE gesichert wurden. Dateien, die mit EXPORT gesichert wurden, müssen mit der Variablen Action ONXIMPORT, siehe Seite **Fehler! Es wurde kein Textmarkenname vergeben.** restauriert werden.

Falls mehrere Sicherungsversionen derselben Datei mit ONXRESTORE zurückgesichert werden, so wird die erste Sicherungsversion unter dem originalen Dateinamen erstellt. Die weiteren Sicherungsversionen werden als Dateien erstellt, wobei dem Originalnamen die Suffix '.$$*nn*' angehängt wird. *nn* ist hierbei eine fortlaufende Nummer welche mit 01 beginnt.

Elemente aus einer Bibliothek in Dateien umwandeln

ONX**SEL** ['*str1*'=...] [, *vsn*/*device*]

Aus den angekreuzten Bibliothekselementen werden Dateien erzeugt.

'*str1*'= ... Namenstransformationsregeln

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

'*str1*'='*str2*'
Der Name der Datei wird gebildet, indem im Namen des Bibliothekselements die Zeichenfolge '*str1*' gesucht und durch '*str2*' ersetzt wird. Definitionsgemäß wird der String '␣' am Ende und der Nullstring '' am Anfang eines jeden Namens gefunden.

'*str1*'=DP[*n*][I]
Delete Prefix: Der Name der Datei wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der links davorstehende Namensteil bzw. die *n* links davorstehenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht.

'*str1*'=DS[*n*][I]
Delete Suffix: Der Name der Datei wird gebildet, indem im ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der nachfolgende Namensteil bzw. die *n* nachfolgenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht. Näheres siehe Seite **Fehler! Es wurde kein Textmarkenname vergeben.**.

Enthält der Name des Bibliothekselements die Zeichenfolge '*str1*' nicht, so wird die Variable Action für dieses Bibliothekselement nicht ausgeführt.

Falls '*str1*'=... nicht angegeben wird, erhält die Datei den gleichen Namen wie das Bibliothekselement. Zum Thema "Selektieren von Bibliothekselementen" siehe auch Seite **Fehler! Es wurde kein Textmarkenname vergeben.**: Action-Code S (Select).

vsn/*device* Bezeichnung und Gerätetyp einer Privatplatte, auf der die erzeugten Dateien abgelegt werden.

Hinweise:

Beim Selektieren eines PLAM-Bibliothekselements in eine Datei werden nur Elementsätze der Satzart 1 berücksichtigt.
Eine Ausnahme bilden Elemente vom Typ R. Hier erzeugt CFS mit Hilfe der LMS-Unterprogrammschnittstelle ISAM-Dateien, die mit CFS (ONXADD) oder LMS (ADDR) wieder in Typ R Elemente umgewandelt werden können.

Für Informationen zu den im Element gespeicherten Satzarten siehe Action-Code F.

PLAM-Elemente vom Typ L können von CFS nicht in Dateien selektiert werden.

Bei Selektion von PLAM-Elementen in Dateien werden Elementtyp und Version nicht zur Bildung des Dateinamens herangezogen. Die Voranstellung eines Prefix im Dateinamen erfolgt durch ONXSEL''='*prefix*'

Beispiele:

`ONXSEL''='ABC.S.'` z.B. TEST --> ABC.S.TEST

`ONXSEL` bzw. `ONXSEL''=''` Dateiname = Elementname. Äquivalent hierzu ist die Verwendung des Action-Codes S.

Dateiattribute verändern (CCS / Cache / SMS)

ONX**SET** CCS=[*name* | *NONE]

Es wird für Dateien oder PLAM-Bibliothekselemente der angegebene Coded-Character-Set Name zugeordnet.
*NONE: Die Verknüpfung zu einem Coded-Character-Set Namen wird gelöscht.

ONX**SET** *param*=*wert*

Es wird eines der neuen Dateiattribute bezüglich Caching (Hiperfile) bzw. SMS (System Managed Storage) gesetzt bzw. verändert. Es kann nur jeweils ein Kennwort zusammen mit dem entsprechenden Parameter angegeben werden.

ADMINFO='*info*' | NO

'*info*' Der Systemverwalter kann im Systemkatalog zu der Datei eine ein bis acht Byte lange Administrations Information *info* hinterlegen (ab OSD V3.0).

NO Es wird keine Administrations Information zu der Datei hinterlegt. Eine evtl. bestehende Information wird gelöscht.

AVAIL=HIGH | STD

HIGH High Availability. Die Datei erhält die Eigenschaft der hohen Verfügbarkeit. Dateien mit hoher Ausfallsicherheit werden vom BS2000 automatisch auf einen entsprechenden Datenträger, z.B. gespiegelte Platten verlagert. Ist kein geeigneter Datenträger verfügbar, wird die Operation mit einem Fehler abgewiesen.

STD Es werden keine besonderen Anforderungen bezüglich Ausfallsicherheit an die Datei gestellt.

DELDATE=*dd.mm.yyyy* | *nnn* | NONE

dd.mm.yyyy Lösch-Freigabedatum für Datei. Die Datei kann ab dem angegebenen Datum ohne Berücksichtigung der Schutzattribute vom Eigentümer gelöscht werden (ab OSD V3.0).

nnn ein- bis fünfstellige Zahl, die die Anzahl der Tage ab dem aktuellen Datum für die Löschfreigabe angibt.

NONE Es wird kein Lösch-Freigabedatum vereinbart. Die Schutzattribute der Datei werden beim Löschen berücksichtigt.

DISKWR=IMMEDIATE | BY-CLOSE

IMMEDIATE Die Daten der Datei müssen sich direkt nach Beendigung der Schreiboperation in einem konsistenten Zustand befinden. Die Datei soll nicht über einen flüchtigen Schreib-Cache bearbeitet werden.

BY-CLOSE Die Daten der Datei müssen sich erst nach der Close-Verarbeitung in einem konsistenten Zustand befinden.

IOPERF=HIGH | VERY-HIGH | STD | USER-MAXIMUM

HIGH Die Datei soll über einen Cache bearbeitet werden (hohe Performance-Priorität)

VERY-HIGH Die Datei soll über einen Cache bearbeitet werden. Falls möglich, soll die gesamte Datei permanent im Cache gehalten werden.

STD Die Datei erhält kein besonderes Performance-Attribut und wird deshalb nicht über einen Cache bearbeitet.

USER-MAX Die Datei erhält das Performance-Attribut, das für den benutzer als höchstes zulässiges in der Joinfile eingetragen ist.

IOUSAGE=READ-WRITE | READ | WRITE

READ-WRITE Ein erhöhtes Performance-Attribut bei der Datei wird für Lese- und Schreiboperationen gewünscht.

READ Ein erhöhtes Performance-Attribut bei der Datei wird nur für Leseoperationen gewünscht.

WRITE Ein erhöhtes Performance-Attribut bei der Datei wird nur für Schreiboperationen gewünscht.

MANCLASS=*class* | NONE

class Gibt eine mit HSMS definierte Management-Klasse an, über die die HSMS-Funktionen Dateisicherung, Archivierung und Langzeitarchvierung gesteuert werden. Die Zuweisung einer Management-Klasse ist nur für Dateien auf SM-Pubsets ab OSD V3.0 möglich.

NONE Eine evtl. definierte Management-Klasse für die Datei wird zurückgesetzt. Für die Dateisicherung gelten die Standardwerte bzw. die mit dem Action-Code MA/MI definierten Verdrängungs-Eigenschaften.

S0MIGR=ALLOWED | FORBIDDEN

ALLOWED Die Datei kann innerhalb der Speicherebene S0 auf einen anderen Volume-Set des gleichen SM-Pubsets verlagert werden. Die Verlagerung kann explizit mit HSMS oder implizit durch das BS2000 erfolgen. Die Zuweisung ist ab OSD V3.0 möglich.

FORBIDDEN Die Datei kann innerhalb eines SM-Pubsets nicht verlagert werden. Diese Zuweisung ist dem Systemverwalter vorbehalten.

STOCLASS=*class* | STD | NONE | UPDATE

class Der Datei auf einem SM-Pubset wird die angegebene Storage-Klasse zugewiesen. Die Namen und Eigenschaften der Storage-Klassen werden durch den Systemverwalter definiert und können durch das BS2000-Kommando /SHOW-STORAGE-CLASS abgefragt werden. Über Storage-Klassen kann der Ablageort von Dateien innerhalb eines SM-Pubsets auf einfache Art geregelt werden. Durch Zuweisung einer Storage-Klasse kann es zu einer impliziten Migration der Datei auf ein anderes Volume-Set innerhalb des SM-Pubsets kommen.
Die Zuweisung einer Storage-Klasse ist ab OSD V3.0 möglich.

STD Der Datei wird die Default Storage-Klasse aus dem Benutzereintrag für den SM-Pubset zugewiesen. Falls keine Default Storage-Klasse definiert ist, erhält die Datei dieselben Attribute wie bei STOCLASS=NONE.

NONE Der Datei wird keine Storage-Klasse zugewiesen. Die Eigenschaften des Ablageorts der Datei wird über folgende Merkmale bestimmt (siehe entsprechende Parameter des Kommandos MODIFY-FILE-ATTR): WORK-FILE, VOLUME-SET, VOLUME, DEVICE-TYPE, IO-ATTRIBUTES, FILE-PREFORMAT, DISK-WRITE, AVAILABILITY.

UPDATE Ist der Datei bereits eine Storage-Klasse zugeordnet, so werden die aktuellen Attribute der Storage-Klasse erneut aus dem Storage-Klassen-Katalog gelesen und der Datei zugewiesen. Diese Option ist dann sinnvoll, wenn sich die Attribute bestimmter Storage-Klassen seit der letztmaligen Zuweisung zu der Datei geändert haben.

USERINFO='*info*' | NO

'*info*' Der Benutzer kann im Systemkatalog zu der Datei eine ein bis acht Byte lange Information *info* hinterlegen (ab OSD V3.0).

NO Es wird keine Benutzerinformation zu der Datei hinterlegt. Eine evtl. bestehende Information wird gelöscht.

WORK=YES | NO | STD

YES Die Datei auf einem SM-Pubset wird im Systemkatalog als Arbeitsdateien gekennzeichnet und kann somit von der Systembetreuung zu einem von ihr festgelegten Zeitpunkt gelöscht werden (ab OSD V3.0).

NO Die Datei soll keine Arbeitsdatei sein.

STD Die Datei soll nur dann Arbeitsdatei sein, wenn sie bereits auf einem Volume-Set für Arbeitsdateien gespeichert ist.

User-Datum für PLAM-Bibliothekselemente verändern

ONX**SETUDAT** [*dd.mm.yyyy*] [*/hh:mm*]]

dd.mm.yyyy Das Feld User-Date aller mit X gekennzeichneten PLAM-Elemente wird auf das angegebene Datum gesetzt.
Falls die Angabe *dd.mm.yyyy* weggelassen wurde, so wird das User-Datum der PLAM-Elemente auf das aktuelle Datum und die aktuelle Uhrzeit gesetzt.

/hh:mm Uhrzeitangabe für das User-Datum.
Falls nur eine Uhrzeit angegeben ist (ONXSETUDAT /hh:mm oder ONXSETUDAT hh:mm), so wird das Datum der PLAM-Elemente unverändert gelassen und es wird nur die Uhrzeit im User Date auf den entsprechenden Wert gesetzt.

Dateien/Bibliothekselemente einzeln oder zusammen sortieren

ONX**SORT** ['*str1*'=...] [SF=*fld*] [, OUT=*datei*] [, ?]
ONX**SORT** ['*str1*'=...] [?] [%*name*] [, OUT=*datei*]

Mit dieser Variablen Action wird das Programm SORT des BS2000 aufgerufen. Die angekreuzten Dateien/Bibliothekselemente können einzeln bzw. zusammen in eine Ausgabedatei sortiert werden.

'*str1*'= ... Namenstransformationsregeln

'*str1*'='*str2*' | '*str1*'=DP[*n*][l] | '*str1*'=DS[*n*][l]

'*str1*'='*str2*'
Die mit dem Action-Code X markierten Datenobjekte werden einzeln sortiert. Die Namen der sortierten Datenobjekte werden gebildet, indem in den ursprünglichen Namen die Zeichenfolge *str1* gesucht und durch *str2* ersetzt wird. Definitionsgemäß wird der String '⊔' am Ende und der Nullstring '' am Anfang eines jeden Namens gefunden. Enthält der Name eines Datenobjekts die Zeichenfolge *str1* nicht, so wird die Sortierung für dieses Objekt nicht ausgeführt.

'*str1*'=DP[*n*][I]
Delete Prefix: Die Namen der neuen Datenobjekte werden gebildet, indem in den ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der links davorstehende Namensteil bzw. die *n* links davorstehenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht.

'*str1*'=DS[*n*][I]
Delete Suffix: Die Namen der neuen Datenobjekte werden gebildet, indem in den ursprünglichen Namen die Zeichenfolge '*str1*' gesucht und der nachfolgende Namensteil bzw. die *n* nachfolgenden Stellen gelöscht werden. Aufgrund des Zusatzes I wird auch das Suchitem '*str1*' gelöscht.

Fehlt der Operand '*str1*'=..., wird je nach Inhalt des Parameters OUT= ein Sort der Dateien in sich selbst (OUT=*SAME) bzw. ein Merge in eine Ausgabedatei durchgeführt (OUT=*datei*).

SF= Sort-Fields.
(*feld-1*) [, (*feld-2*)] .... bzw.
((*feld-1*) ,(*feld-2*) .... ,(*feld-n*)) [,FORMAT=*format*] [,SIZE=*n*] [,OPT=...] ...
Angabe der Sortierfelder und deren Formate und evtl. weiterer Optionen gemäß der Syntax der SORT-Anweisung im Programm SORT. Für eine ausführliche Beschreibung siehe Manual SORT (BS2000). Beispiele für die Schreibweise:

```
(5,10),(20,2,D)
((5,10,A),(20,2,D)),FORMAT=BI,OPT=SEL
```

Bei der Angabe der Sort-Fields ist zu beachten, daß der erste Parameter (Position des Sortierfeldes innerhalb des Satzes) unabhängig von der RECFORM der zu sortierenden Datei stets wie bei einer Datei mit RECFORM=V angegeben werden muß. Deshalb muß die erste Stelle eines Satzes auch bei Recform-F Dateien mit dem Wert 5 bezeichnet werden.

OUT= *datei* Name der sortierten Ausgabedatei. Falls mehrere Eingabedateien angekreuzt wurden, wird in *datei* das Resultat der Sortiervorgangs über alle Eingabedateien gespeichert. Falls *datei* nicht existiert, so wird sie mit FCBTYPE=SAM,RECFORM=V,BLKSIZE=(STD,16) angelegt.

bibl(*element*) Name der Ausgabebibliothek und des Elements, in dem das Ergebnis des Sortiervorgangs abgelegt werden soll.

*SAME Einzelsortierung. Die Datenobjekte werden durch die sortierten überschrieben. '*str1*'=... darf hierbei nicht angegeben werden.

bibl([*t*/]*SAME) Einzelsortieren von Datenobjekten. Die sortierten Objekte werden unter dem ursprünglichen Namen und dem Elementtyp *t* in der angegebenen PLAM-Bibliothek gespeichert. Standard für *t*: D

Der Name einer Ausgabedatei kann auch zu Beginn des CFS-Laufs per File-Kommando festgelegt werden: /FILE *datei*,LINK=SORTOUT,FCBTYPE=..
Bei Angabe einer Namenstransformationsregel '*str1*'=... wird der Parameter OUT= ignoriert.

? Es wird in die Maske zur Eingabe der Sort-Parameter verzweigt. In dieser Maske kann eine Vielzahl weiterer Optionen, Parameter und Exits zur Steuerung des Sortiervorgangs angegeben werden.

%name Parameter-Set, in den zu einem früheren Zeitpunkt alle in der Sort-Parameter-Maske von CFS eingegebenen Parameter gesichert wurden.

?,*%name* wie oben. Die in dem Parameter-Set *%name* gesicherten SORT-Parameter werden am Bildschirm angezeigt und können noch modifiziert werden.

Hinweise:

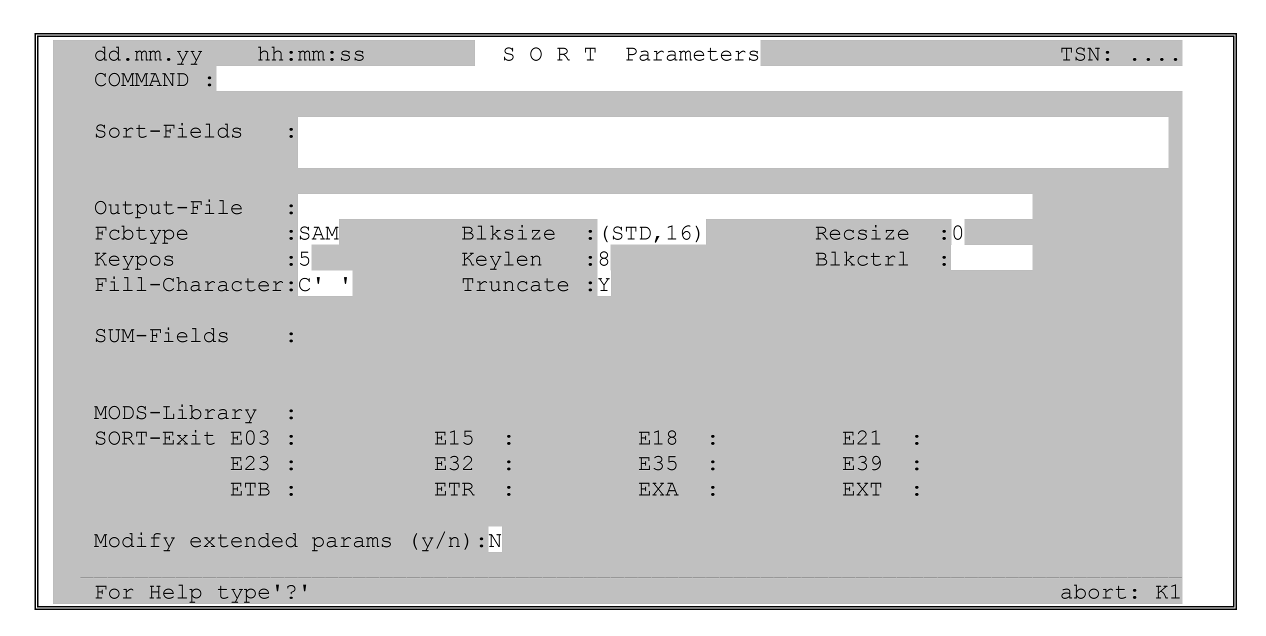
Bei Angabe des Parameter OUT= und Weglassung des Parameters SF= wird von CFS keine Sort-Maske ausgegeben. Die Sort-Anweisungen werden in diesem Fall mit RDATA vom Programm SORT selbst angefordert. Es können alle Sort-Anweisungen, siehe Manual SORT (BS2000), angegeben werden. Als letzte Anweisung ist END anzugeben.

Pamkey- und Adresslistensort werden durch die Variable Action nicht unterstützt.

Bei Eingabedateien mit Recform=V ist zu beachten, daß die Sortierfelder innerhalb der Satzgrenzen aller Eingabedateien liegen.

Datensätze aus Recform-F Eingabedateien werden vor der Übergabe an das Programm SORT auf Recform-V Format gebracht.

Format der Sort-Parameter Maske:



```
dd.mm.yy    hh:mm:ss        S O R T  Parameters                       TSN: ....
COMMAND :

Sort-Fields    :


Output-File    :
Fcbtype        :SAM          Blksize  :(STD,16)        Recsize   :0
Keypos         :5            Keylen   :8               Blkctrl   :
Fill-Character:C' '          Truncate :Y

SUM-Fields     :


MODS-Library   :
SORT-Exit E03 :          E15  :          E18  :          E21  :
          E23 :          E32  :          E35  :          E39  :
          ETB :          ETR  :          EXA  :          EXT  :

Modify extended params (y/n):N

For Help type'?'                                                      abort: K1
```

Beschreibung der Eingabefelder der Sort-Parameter Maske:

COMMAND : HC [*datei*] / NHC   Hardcopy-Modus ein- /ausschalten

/*cmd* beliebiges BS2000-Kommando

SP *%name* die in der Maske eingetragenen Werte werden in einen Parameter-Set mit dem angegebenen Namen gesichert.

LP *%name* die zu einem früheren Zeitpunkt mit SP in einen Parameter-Set gesicherten Sort-Angaben werden in die aktuelle Maske eingetragen. Das Kommando hat die gleiche Wirkung wie die Variable Action ONXSORT ?,*%name*.

? Helpsystem aufrufen.

F3 Aufzeichnung der aktuellen Maske in die Hardcopy-Datei von CFS.

K1 Abbrechen der Variablen Action ONXSORT.

Shift ENTER (DUE2) Aktivierung des Kommandogedächtnisses.

Sort-Fields : Gleiches Eingabeformat wie Schlüsselwortparameter SF=

Output-File : Gleiches Eingabeformat wie Schlüsselwortparameter OUT=

Fcbtype, Blksize, Recsize, Keypos, Keylen, Blkctrl : Eigenschaften der Sort-Ausgabedatei.
Die Angabe Recsize = 0 impliziert Recform=V.
Die Angabe Recsize > 0 impliziert Recform=F.
Bei der Einzelsortierung von Dateien (ONXSORT'*str1*'=...) erhalten die sortierten Dateien die gleiche Recform, Recsize, Keypos, Keylen usw. wie die Eingabedateien, falls die in der SORT-Maske vorgegebenen Standardwerte für Fcbtype, Blksize, Recsize, Keypos, Keylen, Blkctrl unverändert abgesendet werden.

Fill-Character : Falls bei Recsize ein Wert > 0 angegeben wurde, so werden nach der Sortierung kürzere Ausgabesätze mit diesem Zeichen auf die gewünschte Länge aufgefüllt.

Truncate :
- Y Falls bei Recsize ein Wert > 0 angegeben wurde, so werden nach der Sortierung längere Ausgabesätze verkürzt.
- N Falls bei Recsize ein Wert >0 angegeben wurde und nach der Sortierung längere Ausgabesätze vorkommen, so wird SORT mit einem DVS-Fehler abgebrochen.

SUM-Fields : Definition von Summenfeldern. Das Eingabeformat ist das gleiche wie bei den Sort-Fields bzw. beim Schlüsselwortparameter SF=. Für weitere Informationen zu den Summenfeldern siehe Manual SORT (BS2000), Sort-Anweisung SUM.

MODS-Library : Name einer LMR-/PLAM-Bibliothek, in der die angesprochenen Sort-Exits (siehe unten) enthalten sind. Der Name der MODS-Library kann auch zu Beginn des CFS-Laufs per File-Kommando festgelegt werden: /FILE *datei*, LINK=SORTMODS.

SORT-Exit E03, ..., EXT : Benutzerausgänge, bei dessen Erreichen die angegebenen Module aus der MODS-Library aufgerufen werden sollen. Die Angabe eines Moduls *name* für den Sort-Ausgang *xx* entspricht einer Sort-Anweisung MODS E*xx*=(LINK=*name*). Für weitere Informationen siehe Manual SORT (BS2000), Sort-Anweisung MODS.
Bei der Programmierung derartiger Module ist zu beachten, daß stets Datensätze im Recform-V Format übergeben werden.

Modify extended params : Y Es wird in eine zweite Maske verzweigt, in der die Parameter für INCLUDE-, OMIT- und NEWCOL-Anweisungen eingetragen werden können.

Format der Maske für die extended Parameters:

```
dd.mm.yy     hh:mm:ss      Extended  S O R T  Parameters              TSN: ....
COMMAND :


Include :


Omit    :


Newcol  :







For Help type'?'                                                    return: K1
```

Beschreibung der Eingabefelder der Extended Sort-Parameter Maske:

COMMAND :

HC [*datei*]	Hardcopy-Modus einschalten
NHC	Hardcopy-Modus ausschalten
/*cmd*	beliebiges BS2000-Kommando
SP %*name*	die in dieser und in der Hauptmaske eingetragenen Werte werden in einen Parameter-Set mit dem angegebenen Namen gesichert.
?	Helpsystem aufrufen.
F3	Aufzeichnung der aktuellen Maske in die Hardcopy-Datei von CFS.
K1	Abbrechen der Variablen Action ONXSORT.
Shift ENTER (DUE2)	Aktivierung des Kommandogedächtnisses.

Include : Definition der Parameter für eine INCLUDE-Anweisung des SORT. Für weitere Informationen zur Include-Anweisung siehe Manual SORT (BS2000), Sort-Anweisung INCLUDE. Beispiele zur Schreibweise:

```
(7,4,BI,EQ,11,4,BI)
(7,4,BI)=C'&&&&'
```

Omit : Definition der Parameter für eine OMIT-Anweisung des SORT. Für weitere Informationen zur Omit-Anweisung siehe Manual SORT (BS2000), Sort-Anweisung OMIT. Beispiele zur Schreibweise:

```
(7,4,BI,EQ,11,4,BI)
(7,4,BI)=C'&&&&'
```

Newcol : Definition der Parameter für eine Newcol-Anweisung des SORT. Für weitere Informationen zur Newcol-Anweisung siehe Manual SORT (BS2000), Sort-Anweisung NEWCOL. Beispiele zur Schreibweise:

```
(Aa,aÄ,Ää)
(Aa,aÄ,Ää),(00F8,F800)
```

Beispiele:

```
ONXSORT SF=(5,8),(20,3,D),OUT=CFS.SORTDATEI
```

Die Datensätze aller mit X angekreuzten Dateien werden als Gesamtheit sortiert in die sequentielle Datei CFS.SORTDAT ausgegeben. Sätze aus Recform=F Eingabedateien werden vor der Sortierung in V-Format umgewandelt. Das erste für die Sortierung zu verwendende Datenfeld beginnt in Spalte 5, d.h. direkt nach dem vier Byte langen Satzlängenfeld und ist 8 Byte lang. Die Sortierreihenfolge bezüglich dieses

Feldes ist aufsteigend. Das zweite Sortierfeld beginnt in Spalte 20 und umfaßt drei Bytes. Die Sortierreihenfolge bezüglich dieses Feldes ist absteigend.

`ONXSORT' '='.SORT' %SORT1`
Alle mit X angekreuzten Dateien werden als Einzeldateien sortiert. Die Namen der sortierten Dateien werden aus den Eingabedateinamen gebildet, indem die Suffix '.SORT' angehängt wird. Die Sort-Parameter, die Attribute der Ausgabedateien, sowie alle übrigen Angaben zur Steuerung des Sortiervorgangs (Summenfelder, Sort-Exits usw.) werden aus dem Param-Set %SORT1 entnommen.

Dateien mit XPRESS komprimieren/dekomprimieren

ONX**XPRC** ['*str1*'=...] [, MSG|NOMSG] [, *params*] [, INST=$*userid*.]
ONX**XPRD** ['*str1*'=...] [, MSG|NOMSG] [, *params*] [, INST=$*userid*.]

Die mit dem Action-Code X markierten Dateien werden mit dem Programm XPRESS komprimiert (XPRC) bzw. dekomprimiert (XPRD).
Um diese Variable Action anwenden zu können, muß das Softwareprodukt XPRESS verfügbar sein.

'*str1*'= ... Namenstransformationsregeln. Näheres siehe Seite **Fehler! Es wurde kein Textmarkenname vergeben.**.

'*str1*'='*str2*' | '*str1*'=DP[*n*][I] | '*str1*'=DS[*n*][I]

MSG | NOMSG Die von XPRESS bei der Komprimierung erzeugten Meldungen werden am Bildschirm angezeigt/nicht angezeigt. Standard: NOMSG

params wahlweise Parameter zur Steuerung des Unterprogramms XPRESS.
Bei Angabe von mehreren Parametern müssen diese durch Kommas getrennt werden.

INST=$*userid* Benutzerkennung, unter der die XPRESS-Software installiert ist.
Standard: $ST

Beispiel:
`ONXXPRC ' '='.XPR',MSG,CLST,INST=$CFS.`

Dateien in ZIP-Archiv aufnehmen

ONX**ZIP** *zipfile* [, F=C|B] [, M=S|N|TO-E|TO-A] [, C=D|BEST-S|BEST-C] [, PASS=*psw*]

Die mit dem Action-Code X markierten Dateien werden unter gleichem Namen in das angegebene ZIP-Archiv aufgenommen.

zipfile Name des ZIP-Archivs.
Es ist z.Z. noch nicht möglich, die gleiche Datei ein zweites Mal in das selbe ZIP-Archiv aufzunehmen.

F= Format, in dem die Datei in das ZIP-Archiv aufgenommen werden soll.

C (Compatible)
Die so aufgenommene BS2000-Datei kann nach dem Transferieren des ZIP-Archivs in andere Betriebssysteme (mittels FTP im Binärmodus), auch dort noch selektiert und korrekt weiter verarbeitet werden.

B (BS2000-Modus)
Die so aufgenommene BS2000-Datei kann nur in BS2000-Systemen selektiert und korrekt verarbeitet werden.

M= Kodierungsmodus, in dem die Datei im ZIP-Archiv gespeichert wird.

S (Standard)
Die Eingabedateien werden gemäß dem Format des ZIP-Archivs konvertiert. Siehe Parameter F=

N (None)
Die BS2000-Dateien (SAM/ISAM/PAM) werden bei der Aufnahme in das ZIP-Archiv keiner Code-Umwandlung unterworfen.

TO-A (To ASCII)
Im EBCDIC-Code vorliegende BS2000-Eingabedateien (SAM/ISAM) werden bei der Aufnahme in das ZIP-Archiv in ASCII-Code umgewandelt.

TO-E (To EBCDIC)
Im ASCII-Code vorliegende Eingabedateien (SAM/ISAM) werden bei der Aufnahme in das ZIP-Archiv in EBCDIC-Code umgewandelt.

C= Komprimierungsmodus, in dem die Datei im ZIP-Archiv gespeichert wird.

D (Default)
Dieser Modus wählt einen Kompromiss aus Geschwindigkeit und guter Komprimierungsrate.

BEST-C (Best Compression)
Beste Komprimierungsrate zulasten höherer CPU-Belastung.

BEST-S (Best Speed)
Die Komprimierungsrate wird zugunsten geringerer CPU-Belastung reduziert.

PASS=*psw* '*passw*' | L'*passw*' | X'*passw*'

Die in das Archiv aufzunehmenden Elemente werden mit einem Passwort geschützt.

'*passw*' Das Passwort wird in Grossbuchstaben umgewandelt.
L'*passw*' Das Passwort enthält Gross- und Kleinbuchstaben.
X'*passw*' Das Passwort ist hexadezimal angegeben.

Hinweise:

Mit der Variablen Action ONX ZIP können nur Dateien und keine Bibliothekselemente in ein ZIP-Archiv aufgenommen werden. Das ZIP-Archiv wird neu angelegt, falls es noch nicht existiert.

Anstelle von ONX ZIP können Dateien und auch Bibliothekselemente mit **ONX ADD** *zipfile* in ein ZIP-Archiv aufgenommen werden.
Falls die Datei *zipfile* noch nicht existiert, wird der Benutzer gefragt, ob er eine neue Bibliothek oder ein neues ZIP-Archiv anlegen möchte.
Automatisch wird ein ZIP-Archiv erzeugt bei **ONX ADD** *zipfile* ,CLZ (Create Library Z).

Mit ONXADD wird das ZIP-Archiv in einem nur im BS2000 lesbaren Format angelegt. (EBCDIC-Codierung der Daten)

Wird der ZIP-Container mit ONXZIP und dem Standardwert F=C erzeugt, so sind das ZIP-Archiv und die Elemente auch in der offenen Welt lesbar. Dabei ist jedoch folgendes zu beachten: Enthält eine aufzunehmende ISAM-Datei besondere Hexazeichenfolgen, wie z.B. X'0D25' in den Daten, so wird das entspr. Element im BS2000 nicht mehr als Datei selektierbar sein (DMS0AAB, nicht aufsteigende ISAM-Schlüssel), da im ASCII codierten ZIP-Element EBCDIC X'0D25' nach X'0D0A' (CR LF) umgewandelt wird und die Datensätze somit in zwei Teilsätze aufgeteilt werden.

Beispiel:
```
ONXZIP TMP.ZIPFILE,M=BEST-C
```

Dateien aus ZIP-Archiv selektieren

ONX**UNZIP** ['*str1*'=...] [,DATA-TYPE=NOT-SPECIFIED|CHAR|BINARY|SAM-BINARY] [,CHAR-CONV= BY-CONTAINER-FORMAT|NONE|TO-WIN-ANSI|TO-EBCDIC] [,BLOCK-CONTROL= KEEP|IGNORE] [, PASS=*psw*]

Die mit dem Action-Code X markierten Dateien im ZIP-Archiv werden auf Platte selektiert. Diese Variable Action bietet gegenüber ONXSEL einige Zusatzparameter, welche das Format und den Zeichensatz der selektierten Datei bestimmen.

'*str1*'='*str2*' Wahlweise Namenstransformationsregeln

DATA-TYPE Siehe Systemprogramm BS2ZIP, Statement EXTRACT-FILE

CHAR-CONV Siehe Systemprogramm BS2ZIP, Statement EXTRACT-FILE

BLOCK-CONTROL Siehe Systemprogramm BS2ZIP, Statement EXTRACT-FILE

PASS=*psw* '*passw*' | L'*passw*' | X'*passw*'

Die aus dem Archiv zu selektierenden Elemente sind mit einem Passwort geschützt.

'*passw*' Das Passwort wird in Grossbuchstaben umgewandelt.
L'*passw*' Das Passwort enthält Gross- und Kleinbuchstaben.
X'*passw*' Das Passwort ist hexadezimal angegeben.

Hinweis:

Anstelle von ONX UNZIP können gezipte Dateien auch mit
ONXSEL ['*str1*'=...] wieder als Plattendateien selektiert werden.

Beispiel:

```
ONXUNZIP ''='#'
```
Die mit X markierten Dateien im ZIP-Archiv werden unter Voranstellung des Zeichens '#' im Dateinamen als Temporärdateien auf Platte selektiert.

6. Action-Codes

```
dd.mm.yy   hh:mm:ss    HOST: ........   TSN: ....              USER-ID: ........
COMMAND :
LASTP TYP FILENAME = $TSOS                                AGE PASS LACC  ACTION
   30 *I  xxxxxxxxxxxx  . . . . . . . . . . . . . . .     143 *      62 :
   77 *P  xxxxxxxxxxxx  . . . . . . . . . . . . . . .     143 *      62 :
    9 *P  xxxxxxxxxx  . . . . . . . . . . . . . . . .     143 *      62 :
   49 *P  xxx . . . . . . . . . . . . . . . . . . . .       4 . . .   2 :
   61 *I  xxxxxxxxxxxxxxxx  . . . . . . . . . . . . .       4 . . .   2 :
   10 *S  xxxxxxxxxxxxxxxx  . . . . . . . . . . . . .       4 . . .   4 :
   73  S  xxxxxxxxxxxxxxxxxxx . . . . . . . . . . . .       4 . . .   2 :
   29  S  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .       4 . . .   2 :
    0  N  xxxxxxxxxxxxxxxxxxxxx . . . . . . . . . . .         . . .   0 :
  245 *P  xxxxxxxxxxx . . . . . . . . . . . . . . . .       4 . . .   2 :
   49  P  xxxxxx  . . . . . . . . . . . . . . . . . .       2 . . .   0 :
  237 *P  xxxxxxxxxxxxx . . . . . . . . . . . . . . .       4 . . .   4 :
   12  S  xxxxxxxxxxxxxxx . . . . . . . . . . . . . .       8 . . .   7 :
    1  S  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .      45 . . .   4 :
    1  S  xxxxxxxxxxxxxxxxxxxx  . . .   . . . . . . . .    42 . . .   4 :
   14 *P  xxxxx . . . . . . . . . . . . . . . . . . .     290   RW   62 :
    9 *P  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .     148    W   53 :
    9  P  xxxxxxxxxxxxxxxxx . . . . . . . . . . . . .     148 *       1 :
*TAPE  S  xxxxxxxxxxx . . . . . . . . . . . . . . . .       4 . . .   4 :
   11 *P  xxxxxxxxxxxxxxxxxxxxxxx . . . . . . . . . .     148 *ERW   62 :
List continues  (P=1/T=187/H=0)           For Help:  ? / ?cmd / ?all / ??
```

Allgemeine Bemerkungen zu Action-Codes

Action-Codes sind Kürzel aus einem oder zwei Buchstaben, über die in den Action-Feldern der Dateienliste bestimmte Verarbeitungsgänge für die entsprechend markierten Datenobjekte angefordert werden. Die durch Action-Codes angeforderten Verarbeitungen werden im allgemeinen nach Absenden der Maske (ENTER-Taste) ausgeführt.

Einige besondere Action-Codes (z.B. X [Variable Action ausführen], P [Print], E [Erase/ Delete], S [Select]) werden zunächst in einer internen Tabelle gesammelt. Die Ausführung dieser Actions erfolgt erst, wenn das Ende der Dateienliste am Bildschirm angezeigt wurde bzw. wenn der Benutzer die Ausführung explizit mit dem Kommando A [Actions ausführen] verlangt. Das Kommando A hat den Vorteil, daß man bei einer längeren Dateienliste nicht bis zum Ende blättern muß, bevor die gesammelten X-Actions ausgeführt werden.

Hinweis:

Eine vergleichende Gegenüberstellung von Action-Codes und Variablen Actions finden Sie auch auf Seite **107** "Variable Actions".

Benutzerdefinierte Action-Codes

%*act* Es wird die durch den Mnemo-Code *act* definierte benutzereigene Verarbeitung für das angekreuzte Datenobjekt ausgeführt.

%n*act* Die mit dem Mnemo-Code *act* definierte benutzereigene Verarbeitung wird in der Connection n ausgeführt.

% | %? Es wird eine Übersicht aller in der primären und sekundären USERACT-Datei definierten Mnemo-Codes ausgegeben. Die in der sekundären (zentralen) USERACT-Datei enthaltenen Mnemo-Codes sind hierbei durch das Zeichen '*' gekennzeichnet. Für weitere Informationen siehe unten: "Hierarchie der USERACT-Dateien".

Die Zuordnung der Mnemo-Codes %*act* zu den gewünschten benutzereigenen Verarbeitungen erfolgt über die Datei CFS.USERACT. Der formelle Aufbau dieser Datei wird im folgenden beschrieben.

Datei CFS.USERACT für benutzereigene Action-Codes

Mit der Datei CFS.USERACT wird dem CFS-Administrator und jedem einzelnen CFS-Benutzer die Möglichkeit geboten, alphanumerischen Mnemo-Codes (%*act*) individuelle Arbeitsabläufe zuzuordnen. Jeder %-Action-Code kann mit einer BS2000 DO-Prozedur, einer CFS-Prozedur oder einem beliebigen anderen CFS-Kommando verknüpft werden.

Die Datei CFS.USERACT ist eine gewöhnliche mit EDT erstellbare SAM-Datei, die 3 verschiedene Arten von Sätzen kennt:

- Kommentarsätze können an beliebiger Stelle eingestreut werden und beginnen in Spalte 1 mit einem Stern ('*').

- Header-Sätze beginnen in Spalte 1 mit '$'. Ab Spalte 2 folgt

  der String '%', gefolgt von einem beliebigen CFS-Kommando. Es kann z.B. ein E-Kommando zum Starten einer DO-Prozedur im Enter über das Job Report-System angegeben werden. Dem Kommando können über Kurzbezeichnungen (siehe unten) die Namen der zu bearbeitenden Datenobjekte mitgegeben werden.
  Das anzugebende CFS-Kommando kann in einer oder mehreren Fortsetzungszeilen weitergeführt werden. Hierzu ist am Ende der fortzusetzenden Zeile das Zeichen '-' anzugeben. Die Fortsetzungszeile beginnt mit '$', einem oder mehreren Blanks und dem fortzusetzenden Text. Ein Beispiel hierfür ist weiter unten aufgeführt.

  Die Namen der zu bearbeitenden Datenobjekte können über die folgenden Kurzbezeichnungen in das Kommando eingefügt werden:

!NAME | !FILE	Name der Datei/Jobvariablen
!ELEMENT	Bezeichnung des Bibliothekselements ohne Typ und Version
!LIBRARY	Name der Bibliothek, in der das Element enthalten ist
!TYP	Typ des Bibliothekselements bzw. Fcbtyp bei Dateien
!VERS	Version des Bibliothekselements
!CFN	Bei Dateien und JV's: Name des Datenobjekts (wie !NAME, !FILE) Bei Bibliothekselementen: bib-name(t/element/vers)
!RFN	Bei Dateien und JV's: Name ohne Cat-Id und User-Id.

Hinweise:

Ein Header-Satz kann bis zu 2048 Bytes lang sein. Falls die USERACT-Datei mit EDT bearbeitbar sein soll, ist die maximale Satzlänge auf 256 Bytes begrenzt.

Falls in einem Header-Satz das Kommando E *do-proc* angegeben ist (z.B. $%E CFS.S.LMSLIB(ASSEMB)), wird jedes der mit dem Action-Code %*act* (z.B. %ASS) markierten Datenobjekte unter Überwachung des Job-Report Systems in einem eigenen Enter-Auftrag bearbeitet. Durch %*act*? wird die Ausgabe der DO-Parameter Maske für die Prozedur erzwungen. Falls in CFS bereits eine Connection mit der Nummer *n* zu $DIALOG eröffnet ist (z.B. OC1/LOGON ...), kann durch Angabe des Action-Codes %*nact* die Verarbeitung in der angegebenen Connection im Dialog erfolgen (z.B. %1ASS).

Falls in einem Header-Satz das Kommando FT ohne eine Kurzbezeichnung für Datei- oder Elementnamen wie !NAME, !ELEMENT, usw. angegeben ist (z.B. $%FT ,,,HOST22,TEST2,A001,,,,*BS2000,TO,NO), so wird bei der Ausführung des Kommandos der Name des mit FT zu übertragenden Datenobjekts ergänzt. Die gleiche Funktionalität ist auch über den Action-Code PD (siehe CFS.PDFILE) realisiert. Im Unterschied zu %*act* sind bei PD*mn* jedoch nur ein- bis zweistellige Mnemo-Codes *mn* möglich.

Auf einen Header-Satz folgen ein oder mehrere Mnemo-Code Sätze (%*act*).

- Mnemo-Code Sätze beginnen in Spalte 1 mit einem alphabetischen Zeichen. Die Felder haben folgendes Format (siehe auch Beispiel weiter unten):

  Spalte 1-4 — Variabler Teil *act* des Action-Codes %*act* (*act*: alphanumerisch, linksbündig, erstes Zeichen ein Buchstabe).

  Spalte 11-80 — beliebiger Text (Beschreibung des mit dem Action-Code %*act* ausgeführten Kommandos bzw. Blank. Die gesamte Liste aller Beschreibungen wird durch Angabe des Action-Codes % bzw. %? am Bildschirm angezeigt.

Zuordnungsalgorithmus: %*act* <--> erzeugtes Kommando:

Es wird der Mnemo-Code *act* in der primären USERACT-Datei gesucht. Bezüglich der Suchreihenfolge bei der Ermittlung der primären USERACT-Datei siehe "Hierarchie der USERACT-Dateien", Punkt 1) bis 3) weiter unten.
Wird der Mnemo-Code *act* in der primären USERACT-Datei nicht gefunden, so wird in der sekundären USERACT-Datei $*user-id*.CFS.USERACT (siehe unten, Punkt 4)) weiter gesucht.

Falls *act* mit einer Ziffer beginnt, wird diese als Connection-Nummer und der Rest als Mnemo-Code interpretiert. Das dem Mnemo-Code zugeordnete Kommando wird in der angegebenen Connection zur Ausführung gebracht.

Falls ein Satz mit der angegebenen Bezeichnung gefunden wurde, wird der vorausgegangene Header-Satz ($...) zur Ermittlung des auszuführenden Kommandos gesucht. Die im Kommando des Header-Satzes enthaltenen Kurzbezeichnungen !NAME, !ELEMENT, !LIBRARY, !TYP, !VERS (siehe oben) werden durch die entsprechenden Namensbestandteile des angekreuzten Datenobjekts ersetzt.

Durch den Action-Code % (bzw. %?) erhält der Benutzer eine Liste aller definierten Mnemo-Codes und deren Zuordnung zu den entsprechenden Kommandos.

Hierarchie der USERACT-Dateien

Die von CFS verwendete USERACT-Datei wird nach der folgenden Hierarchie bestimmt:

1) Existiert ein File-Kommando mit LINK=USERACT (/FILE ...,LINK=USERACT) ?

2) Existiert kein TFT-Eintrag mit dem Namen USERACT, so wird der im LOGON-Kommando angegebene Job-Name (/*.jobname* LOGON ...) an das Prefix 'CFS.USERACT.' angehängt. Die Datei CFS.USERACT.*jobname* wird zuerst in der eigenen Benutzerkennung und falls dort nicht vorhanden, in der CFS-Installationskennung (siehe 4)) gesucht.

3) Wurde beim LOGON-Kommando kein Jobname angegeben oder existiert die Datei CFS.USERACT.*name* nicht, so wird eine Datei CFS.USERACT unter der eigenen Benutzerkennung gesucht.

4) Existiert keine der oben genannten Dateien, so sucht CFS als letztes nach einer Datei $*user-id*.CFS.USERACT. $*user-id* ist die Benutzerkennung, unter der das Programm CFS gespeichert ist (CFS-Installationskennung).

Beispiel für eine Datei CFS.USERACT:

```
$%E !LIBRARY(ASS),!ELEMENT
ASS       Übersetzen eines Bibliothekselements mit der Prozedur ASS in gleicher Bibl
$%E $RZ.SYSPROC(COB85),!NAME
COB       Compilieren einer Datei mit der DO-Prozedur COB85 in $RZ.SYSPROC
$%FT ,,,HOST11,TEST2,A001,,,,*BS2000,TO,NO
F11       Übertragung per File-Transfer auf V11-Host
$%/PRINT !FILE,FORM=L,DEVICE=DR01,-
$        SPOOL-OUT-OPTIONS=*PAR(...............................................)
PR        Langes Print-Kommando mit Spool-Options in Fortsetzungszeile
$%FT ,,,HOST11,TEST2,A001,,,,*BS2000,TO,NO
F11       Übertragung per File-Transfer auf V11-Host
$%EDOR,!NAME
EDOR      Bearbeiten der Datei mit CFS-Kommando EDOR
$%DO PROC1,!NAME
P1
1234      <11 ------------------------------------------------------------------
2048>
```

```
|             |
|             ▫-- Beliebiger Kommentar. Dieser wird bei Eingabe der Action-Codes % / %?
|                 bei dem entsprechenden Mnemo-Code angezeigt.
|                 Falls zu einem Mnemo-Code kein Kommentar eingetragen ist, wird in der
|                 Übersichtsmaske das zu dem Mnemo-Code gehörige Kommando (Inhalt
|                 des Header-Satzes) angezeigt.
|
|
▫---------------- maximal 4 Stellen Mnemo-Bezeichnung des Action-Codes %act
```

Ein weiteres Beispiel einer USERACT-Datei ist in der Bibliothek CFS.S.LMSLIB enthalten.

Hilfe zu Action-Codes anfordern

? Help-Funktion (Liste der verfügbaren Action-Codes anzeigen).
Für eine ausführliche Beschreibung des Help-Systems siehe Seite 547 und folgende.

Zeile in Dateienliste unsichtbar machen

- Element der Dateienliste unsichtbar machen. Der entsprechende Eintrag wird fortan nicht mehr angezeigt. Variable Actions der Form ON& ... werden auf durch '-' markierte Dateien nicht angewendet.
CFS verwendet diesen Action-Code auch intern zum Ausblenden von zuvor in der Dateienliste enthaltenen Datenobjekte, die mit dem Action-Code E (Erase) gelöscht wurden. Mit dem Kommando YANK können alle unsichtbaren Einträge wieder sichtbar gemacht werden.

Sichtfenster auf Zeile in Dateienliste positionieren

+P | + Die Zeile der Dateienliste wird am Bildschirm als erste Zeile angezeigt.

-P Die Zeile der Dateienliste wird am Bildschirm als letzte Zeile angezeigt.

Namen für spätere Verwendung im Kommandofeld merken

***** Das Zeichen * kann im Feld COMMAND von CFS als Platzhalter für den Namen eines in der Dateienliste aufgeführten Datenobjekts verwendet werden. In der Action-Spalte der Dateienliste ist dazu bei der gewünschten Datei der Action-Code * einzutragen.

******n* Durch Angabe einer Ziffer *n* (1 <= *n* <= 9) im Anschluß an den Stern können weitere Kurzbezeichnungen für Namen definiert werden.

Hinweise:

Die Zuordnungen von Datenobjektnamen zu den Action-Codes *1, *2, ...,*9 bleiben während eines CFS-Laufs solange bestehen, bis sie durch neue Zuweisungen ersetzt werden oder mit dem CFS-Kommando CL* gesamtheitlich gelöscht werden. Das Bestehenbleiben der Namenszuordnungen gilt auch nach Selektion einer anderen Dateienliste.

Für die Zuordnung von * zu dem Namen eines Datenobjekts gilt folgendes:
Standardmäßig wird die Zuordnung beliebig lange gemerkt wie bei *1, *2, ... (s.o.). Durch eine Modifikation im CFS-Vorschaltprogramm CFSMAIN kann jedoch der folgende alternative Modus für * aktiviert werden:
Die Namenszuordnung zu * wird gelöscht, nachdem sie im Kommandofeld von CFS zur Namenssubstitution verwendet wurde. Damit werden die Fälle von möglichen Fehlsubstitutionen von Zeichenfolgen wie *any, *priv, *std, *lib usw. im Kommandofeld verhindert.

Falls in der Dateienliste die Elemente einer PLAM-Bibliothek angezeigt werden, wird durch Angabe von * in Kommandos der Bezug zu einem Element in der Form *bibl*(*name*) hergestellt. In bestimmten Fällen ist es jedoch wünschenswert, bei der Namenssubstitution den Bibliotheksnamen zu ignorieren. In diesen Fällen ist im Kommando *E anstelle von * anzugeben.

In BS2000-Kommandos der Art /FSTAT *XYZ*, eingegeben im Kommandofeld von CFS, ist die Ersetzung von * durch einen früher markierten Datei- oder Elementnamen

eine häufig auftretende Fehlerquelle. Um zu verhindern, daß der Stern im Kommando durch einen Namen ersetzt wird, kann das Kommando **CL*** (CLear *) verwendet werden. Dieses Kommando bewirkt, daß die Zuordnung * <--> Name eines Datenobjekts gelöscht wird. * kann fortan in Kommandos eingegeben werden, ohne daß eine Ersetzung durch einen Dateinamen stattfinden würde.

Beispiele:

`NPLMS *,TEST`
In der Dateienliste werden alle Elemente der durch * markierten LMS-Bibliothek angezeigt, die die Zeichenfolge TEST in ihrem Namen enthalten.

`/CAT *,RETPD=0`
Bei der mit dem Action-Code * markierten Datei wird die Retention-Period auf 0 Tage gesetzt. Der im CAT-Kommando nicht angegebene Parameter STATE=U wird von CFS ergänzt.

`E *,TIME=100`
Enter-Auftrag durch das Job-Report System starten. Falls in der Dateienliste die Elemente einer Bibliothek, z.B. der Bibliothek PROC.ENTER, angezeigt werden und das Element J/TSOSLNK mit * markiert wurde, so wird in dem Kommando das Zeichen * durch PROC.ENTER(TSOSLNK) ersetzt.

ACCESS=Read/Write für Datei / Jobvariable

AR | **AW** Access Read / Write für die entsprechend markierte Datei/Jobvariable.
AWE Access Write und Action-Code E (Erase) eintragen.

Availability Status für Datei verändern

AVH | **AVS** Availability High / Standard. Anforderungen bezüglich der Ausfallsicherheit der Datei ändern (ab OSD V3.0).
Dateien mit hoher Ausfallsicherheit (AVH) werden vom BS2000 automatisch auf einen entsprechenden Datenträger, z.B. gespiegelte Platten verlagert. Ist kein geeigneter Datenträger verfügbar, wird der Action-Code mit einem Fehler abgewiesen.

Zugriffsrechte (Basic Access Control List) für Datei/Jobvariable/ PLAM-Bibliothekselement ändern

Vorbemerkungen:

Die im folgenden für Dateien beschriebenen Action-Codes zur Veränderung der Basic Access Control Liste können in gleicher Weise auch auf Jobvariablen und Elemente einer PLAM-Bibliothek angewendet werden. Für ein Element einer PLAM-Bibliothek gilt als User (Owner) die Benutzerkennung unter der die Bibliothek katalogisiert ist.

Die Einrichtung der BACL's steht ab BS2000 V10 zur Verfügung. Bei PLAM-Bibliotheken ist eine LMS-Version >= 2.0 erforderlich.

Bezüglich der Standard-Schutzattribute (Zugriffsrechte) für Elemente einer PLAM-Bibliothek wird auf den Action-Code SP, Seite 204 verwiesen.

AM Access Modify.

Dieser Action-Code setzt voraus, daß bei der Selektion der Dateienliste die User Option BACL[F] angegeben wurde.
Die im Listeintrag dargestellten BACL-Attribute werden überschreibbar und können direkt modifiziert werden. Um eine Zugriffsart für eine bestimmte Benutzerdomäne zu verbieten, ist an der entsprechenden Stelle das Zeichen '-' einzutragen. Um einen Zugriff zu erlauben, ist die Bezeichnung der Domäne (UGO) bzw. die Art des Zugriffs (RWX) einzutragen.

AA[*x*] Access All.

x U | G | O:
Alle Zugriffsarten (d.h. Read/Write/Exec) werden für die angegebene Domäne erlaubt. Für *x* kann eine beliebige Kombination der Buchstaben U, G, O angegeben werden. Z.B. bewirkt der Action-Code AAUGO, daß alle Benutzer uneingeschränkten Zugriff auf die Datei haben.

x W | R | X:
Allen Domänen (d.h. User/Group/Others) wird die angegebene Zugriffsart erlaubt. Für *x* kann eine beliebige Kombination der Buchstaben W, R, X angegeben werden.

Der Action-Code AA ohne einen Zusatz *x* bewirkt, daß die Datei von allen Domänen ohne Einschränkung bearbeitet werden kann.

AN*x* Access No.

x U | G | O:
Alle Zugriffsarten (d.h. Read/Write/Exec) werden für die angegebene Domäne gesperrt. Der Eigentümer der Datei (User, Owner) kann diese Einstellung jederzeit wieder ändern. Für *x* kann eine beliebige Kombination der Buchstaben U, G, O angegeben werden.

x W | R | X:
Für alle Domänen (d.h. User/Group/Others) wird die angegebene Zugriffsart gesperrt. Für *x* kann auch eine beliebige Kombination der Buchstaben W, R, X angegeben werden.

AN No Basic Access Control List. Beim Zugriff auf die Datei sind die BACL-Prüfungen nicht aktiv. Eine Standard Basic Access Control List kann z.B. durch Eingabe des Action-Codes AS wieder aktiviert werden.

AS Access Rights = Standard. Es werden Standard BACL-Attribute für eine noch nicht BACL-geschützte Datei vergeben. Die Standard-Attribute sind abhängig von den bisherigen Werten für Share und Access.

AW*x* | **AR***x* | **AX***x*
Schreibenden/Lesenden/Ausführenden Zugriff für eine bestimmte Gruppe von Benutzerkennungen zulassen.

x Domäne, für die die angegebene Zugriffsart erlaubt werden soll.

- U User. Der Eigentümer (User-ID, unter der die Datei katalogisiert ist) hat Zugriff auf die Datei.
- G Group. Eine festgelegte Menge von Benutzerkennungen (Benutzergruppe) hat Zugriff auf die Datei.
- O Others. Alle Kennungen, die nicht Eigentümer der Datei sind oder der Benutzergruppe des Eigentümers angehören, haben Zugriff auf die Datei.

N None. Weder User, noch Group, noch Others, d.h. niemand hat Zugriff auf die Datei. Der Eigentümer der Datei (U) kann diese Einstellung jedoch jederzeit wieder verändern.

Für *x* kann eine beliebige Kombination der Buchstaben U, G, O angegeben werden. Z.B. bewirkt der Action-Code AXUGO, daß Benutzer unter User, Group und Others ein EXEC auf die Datei ausführen können.

AWN*x* | **ARN***x* | **AXN***x*

Verbot des schreibenden/lesenden/ausführenden Zugriffs für eine Gruppe von Benutzerkennungen.

x Domäne, für die die angegebene Zugriffsart verboten werden soll. Beschreibung siehe oben. Für *x* kann auch eine Kombination der Buchstaben U, G, O angegeben werden. Z.B. bewirkt der Action-Code AWNGO, daß Benutzer aus der Gruppe und allen anderen Kennungen (Others) die Datei nicht verändern dürfen.

Hinweise:

Falls mit dem Action-Code A.. eine Basic Access Control Liste oder ein Passwort für ein Element einer PLAM-Bibliothek definiert oder verändert wurde, so werden für diese Bibliothek auf Dateiebene folgende zusätzlichen Prüfungen durchgeführt: Besitzt die Datei das Attribut SHARE=YES ?. Wenn Ja: Ist für die Datei eine Basic Access Control Liste (BACL) oder eine Full Access Control Liste (FACL) definiert ? Falls Nein, werden für die Datei BACL-Attribute definiert und auf OWNER/GROUP/OTHERS =READ/WRITE/EXEC gesetzt. Damit kann die Datei mit dem Copy-Kommando nicht auf fremde Benutzerkennungen kopiert werden.

Für weitere Informationen zu BACL siehe Seite 78.

Passwort zum Zugriff auf Datei/PLAM-Bibliothekselement definieren

Vorbemerkung: Die im folgenden beschriebene Möglichkeit, READ-/WRITE-/EXEC-Passworte für Elemente einer PLAM-Bibliothek zu definieren, setzt eine LMS-Version >= 2.0 und BS2000 >= 10.0 voraus.

ARP | **AWP** | **AXP**

Es wird eine Maske angefordert, in der der Benutzer ein bis zu vier Byte langes Read-, Write- oder Exec-Passwort für das betreffende Datenobjekt (Datei/Bibliothekselement) vergeben kann. Das Eingabeformat für das Passwort ist [C]'...' bzw. X'...'.

Da die Passworte für Bibliothekselemente in verschlüsselter Form gespeichert werden, kann der Wert eines Passworts im Nachhinein weder vom Systemverwalter, noch vom Eigentümer der Bibliothek ermittelt oder abgefragt werden. Für nicht mehr bekannte Passworte bleibt für den Eigentümer der Bibliothek nur die Möglichkeit, diese zu entfernen und evtl. neu zu vergeben.

ARPN | **AWPN** | **AXPN**

Der Eigentümer, d.h. ein Benutzer, der unter der User-ID angeloggt ist, unter der die Bibliothek katalogisiert ist, kann das Passwort für ein Bibliothekselement entfernen. Es ist hierzu nicht die Kenntnis des betreffenden Passworts notwendig. Bei der Anwendung des Action-Codes auf Dateien, müssen die benötigten Passworte zuvor per PASSWORD-Kommando gesetzt worden sein.

Backup-Level von Dateien verändern

BL*x* Der Backup-Level der Datei wird auf den angegebenen Wert gesetzt.

x Backup-Level: A | B | C | D | E

Datei / Bibliothekselement / ZIP-Element / Jobvariable kopieren

C Copy. Der Name des durch den Action-Code C markierten Datenobjekts wird in der Maske hell dargestellt und kann vom Benutzer überschrieben werden mit dem Namen, den die Kopie des Datenobjekts erhalten soll. Das ursprüngliche Datenobjekt wird dadurch nicht verändert. Der Namenseintrag in der Dateienliste wird nach der Copy-Operation wieder mit dem ursprünglichen Inhalt angezeigt.
Falls eine Datei mit BLKCTRL=PAMKEY auf einen NK-Pubset kopiert werden soll, ruft CFS intern das Dienstprogramm PAMCONV auf.

Eine ordnungsgemäße Ausführung der Copy-Operation wird durch die Quittung C---C ganz rechts im entsprechenden Listeintrag angezeigt. Im Fehlerfall wird hier der DVS- oder Bibliotheks-Fehlercode eingetragen.

Kopierte Datenobjekte werden entweder am Ende der Dateienliste hinzugefügt (SET COPYMODE=E, Standard) oder direkt hinter dem originalen Datenobjekt eingefügt (SET COPYMODE=A).

Ein in Delta-Form gespeichertes Element einer PLAM-Bibliothek wird normalerweise in der Kopie als Voll-Element abgelegt. Ausnahme: Das Zielelement des Kopiervorgangs liegt bereits in delta-gespeicherter Form vor.
Ein nicht delta-gespeichertes PLAM-Element wird auch in der Kopie als nicht delta-gespeichert angelegt. Ausnahme: Das Zielelement des Kopiervorgangs liegt bereits in delta-gespeicherter Form vor. In diesem Fall wird die Version um 1 erhöht.

CA Copy All. Bei einem deltagespeicherten Element einer PLAM-Bibliothek werden alle zugehörigen Versionen strukturerhaltend in ein Element mit dem angegebenen neuen Namen kopiert.

CS Copy,SAME. Dateien/Jobvariable/PLAM-Elemente werden mit den Schutzattributen kopiert. Diese bestehen bei Dateien und Jobvariablen aus den Eigenschaften ACCESS=READ/WRITE, RDPASS/WRPASS/EXPASS, RETPD, Basic Access Control List (BACL). Bei PLAM-Elementen werden die Read-, Write- und Exec-Passworte sowie die Basic Access Control Liste (BACL) in das neue Element übernommen. Bei dem Action-Code C ohne den Zusatz S werden die Schutzattribute der Datenobjekte nicht mit kopiert. Kopien von shareable Dateien werden dann z.B. als nonshare-Dateien angelegt.

Hinweise:

Unter Benutzerkennungen ungleich TSOS können beim Kopieren von Jobvariablen mit dem Action-Code CS (Copy Same) eventuell vorhandene READ- und WRITE-Passworte nicht übernommen werden.

Soll eine Datei aus einer fremden Kennung in die eigene kopiert werden, so muß im Namen der neuen Datei die eigene Benutzerkennung $*user-id.* vorangestellt werden.

Inhalt einer Datei / Jobvariablen / Bibliothekselement anzeigen

D Display. Anzeigen des Inhalts eines Datenobjekts (Datei/Jobvariable/Bibliotheks-element).

Mit dem Action-Code D wird das CFS-Display/Editor-System aufgerufen und das Datenobjekt im Lesemodus eröffnet.
Nach Eintragen des Kommandos M (Modify) kann das Datenobjekt geändert werden.
Das Datenobjekt bleibt solange im Update-Modus eröffnet, bis
a) das Kommando NM (No Modify) eingegeben wird
b) mit dem Kommando NF (Next File) die nächste Display-Datei angezeigt wird
c) der Editor mit der K1-Taste ganz verlassen wird.

Soll das Datenobjekt sofort im Änderungsmodus eröffnet werden, so ist der Action-Code M (Modify) anzugeben.

Für eine ausführliche Beschreibung des CFS-Display und Editor-Systems wird auf Seite 289 und folgende "CFS-Display/Editor" verwiesen.

Hinweise:

Beim Display für sehr große Bibliothekselemente kann es zu Platzproblemen im PUBLIC-Speicher kommen, da CFS für das Element eine temporäre Datei mit dem Namen #Z[*.typ*] [*.elementname*] anlegt.

Display-Actions werden sofort ausgeführt in der Reihenfolge der Eingabe. Um von der Anzeige einer Datei zur Anzeige der nächsten Datei überzugehen, ist im Kommandofeld von CFS der Befehl NF (Next File) bzw. D einzugeben. Mit dem Befehl LF (Last File) kann wieder zur Anzeige der vorhergehenden Datei zurückverzweigt werden. Möchte der Benutzer aus dem Display-Modus in die Dateienliste zurückkehren, ist das Kommando LST (Liste) anzugeben bzw. die K1-Taste auszulösen.

Lösch-Freigabedatum für Dateien verändern

DD*n* Das Lösch-Freigabedatum der Datei wird auf *n* Tage (0 <= *n* <= 999) ab dem heutigen Datum gesetzt. Die Datei kann ab diesem Datum vom Eigentümer ohne Berücksichtigung der Schutzattribute gelöscht werden.

Nicht benötigten Speicherplatz freigeben

DS Deallocate Space. Bei Dateien wird der nicht benötigte Speicherplatz (zugewiesene, jedoch nicht beschriebene PAM-Seiten) freigegeben: FILE ..., SPACE=-99999.

DS [*n*] Deallocate Space. Die nicht benutzten PAM-Seiten der Datei werden bis zu einem Vielfachen von *n* Seiten (0 <= *n* <= 999) freigegeben.

Bei Dateien, die noch nicht eröffnet wurden (nur /FILE-Kommando) werden mit DS alle Seiten bis auf eine minimale Allokation auf der Platte freigegeben.
Mit DS0 werden alle von der Datei belegten Seiten freigegeben. Die Datei hat danach keine Speicherseiten mehr belegt.

DSI Wie DS, jedoch werden evtl. vorhandene Dateiattribute Access=Read und Retpd vor dem DS auf die Standardwerte und danach wieder auf die ursprünglichen Werte gesetzt.
Bei Dateien unter TSOS, deren Namen mit 'SYS' beginnen (z.B. SYS.PAGING. ...) wird das temporäre Zurücksetzen der Attribute aus Sicherheitsgründen nicht ausgeführt.

Datei / Bibliotheks- / ZIP-Element / Jobvariable löschen

E Das Datenobjekt (Datei/Bibliotheks- bzw. ZIP-Element/Jobvariable) wird gelöscht

EA FGG: Alle Generationen nach der markierten werden gelöscht (ERASE ,POS=AFTER)

EA PLAM: alle zu dem Element gehörenden Delta-Versionen werden gelöscht

EB FGG: Alle Generationen vor der markierten werden gelöscht (ERASE ,POS=BEFORE)

EC nur der Katalogeintrag der Datei wird gelöscht (ERASE ,CATALOG)

ED Die Datei / das PLAM-Element wird gelöscht. Der belegte Speicherplatz des Datenobjekts wird aus mit binären Nullen überschrieben (ERASE ..., DESTROY)

EDA Es wird nur der logische Inhalt der Datei gelöscht, d.h. die Highest Used Page wird auf 0 gesetzt. Die Dateiattribute Fcbtype, Recform, Recsize, Blksize usw. werden ebenfalls zurückgesetzt. Der belegte Speicherplatz der Datei sowie der Katalogeintrag bleiben erhalten (ERASE ,DATA)

EDAR Wie Action-Code EDA, jedoch werden die Dateiattribute Fcbtype, Recform, Recsize, Blksize usw. nicht zurückgesetzt.

EN Das Datenobjekt wird ohne Zurückbehalten einer temporären Datei gelöscht. Für weitere Informationen zu der Erase Retain Tempfile Option (ERT) siehe Hinweise weiter unten.

ET Das Datenobjekt wird unter Einschaltung der Erase Retain Tempfile Option (ERT) gelöscht. Für weitere Informationen zu der Erase Retain Tempfile Option (ERT) siehe Hinweise weiter unten.

EI Beim Löschen werden die Attribute ACCESS=READ und RETPD>0 sowie die Basic Access Control Liste ignoriert (ERASE ,IGNORE=(ACCESS,EXDATE)). Unter TSOS werden auch evtl. vorhandene Dateipassworte ignoriert (ERASE ,IGNORE=(ACCESS, EXDATE,RDP,WRP)).

Hinweise:

Beim Löschen von Dateigenerationen durch den Action-Code E wird der Benutzer nach dem POS-Operanden gefragt.
A POS=AFTER alle Dateigenerationen nach der aktuellen werden gelöscht.
B POS=BEFORE alle Dateigenerationen vor der aktuellen werden gelöscht.
N POS= es werden keine Dateigenerationen gelöscht.

Im Zusammenhang mit dem Löschen von Datenobjekten wird auf den in CFS standardmäßig eingeschalteten ERT-Modus hingewiesen.
Der ERT-Modus (Erase with Retain Tempfiles) hat zur Folge, daß die mit dem Action-Code E gelöschten Dateien zunächst in temporäre Dateien umkatalogisiert werden. Diese temporären Dateien werden vom BS2000 bei Prozeßende gelöscht. Dateien, die versehentlich mit dem Action-Code E gelöscht wurden, sind damit noch bis Proze-

ßende physisch als Dateien vorhanden (zugreifbar durch Kommando NP# bzw. NP STAJV #).

Die Namen der erzeugten Temporärdateien werden gebildet, indem beim CAT-Kommando dem Namen der zu löschenden Dateien das Tempfile-Zeichen # vorangestellt wird.

Bibliothekselemente werden vor dem Löschen unter dem gleichen Namen in eine temporäre Bibliothek kopiert und sind dort bis Prozeßende noch zugreifbar. Der Name der temporären Bibliothek wird gebildet, indem dem Namen der Bibliothek, aus der Elemente gelöscht werden, das Tempfile-Zeichen # vorangestellt wird.

Die ERT-Option **wirkt nicht**, falls

- Elemente eines ZIP-Archives gelöscht werden,
- Dateien mit den Action-Codes ED (Erase...,Data), EC (Erase..., Catalog) oder EN (Erase No Retain Tempfiles) gelöscht werden,
- Dateien auf Privatplatten mit mehr als 5000 beschriebenen Seiten gelöscht werden,
- Dateien mit einer PVS-Bezeichnung im Feld USER-ID selektiert wurden,
- unter TSOS Dateien einer fremden Kennung selektiert wurden,
- PLAM-Bibliothekselemente mit dem Action-Code EA (Erase All delta versions) gelöscht werden.

Die ERT-Option läßt sich ausschalten durch das Kommando NERT (No Erase Retain Temporary) bzw. über die SET-Maske.

Bei ausgeschaltetem ERT-Modus können einzelne Dateien dennoch mit der Tempfile-Option gelöscht werden, indem anstelle von E, der Action-Code ET (Erase with Tempfiles) angegeben wird.

Gelöschte Datenobjekte werden automatisch aus der aktuellen Dateienliste entfernt (impliziter Action-Code '-'). Mit dem Kommando YANK können diese unsichtbaren Zeilen der Dateienliste wieder sichtbar gemacht werden.

Bei delta-gespeicherten Elementen einer PLAM-Bibliothek bewirkt der Action-Code E das Löschen der angezeigten Version. Alle übrigen Versionen des Elements werden nicht gelöscht. In der Dateienliste erscheint das Element nach dem Löschen mit der nächst niedrigeren Versionsnummer.
Zum Löschen aller Versionen eines Elements ist der Action-Code EA (Erase All) zu verwenden.

Datei / Bibliothekselement / Jobvariable im EDT bearbeiten

EDT[*n*]

Datei/Bibliothekselement/Jobvariable im Formatmodus des EDT bearbeiten.
Rückkehr in die CFS-Maske mit der **K1**-Taste oder dem EDT-Kommando H [ALT].
Die Action-Codes EDT und EDF sind gleichwertig und bewirken beide den Aufruf des EDT im Formatmodus.

Hinweis:

Der optionale Zusatz *n* steht bei den im folgenden beschriebenen Action-Codes EDT/EDF/UPD für die Nummer einer EDT-Ebene. Das angesprochene Datenobjekt wird in

die angegebene EDT-Ebene *n* eingelesen bzw. aus dieser Ebene zurückgeschrieben. Falls *n* nicht angegeben wurde, gilt die EDT-Ebene 0 als Standard.

EDL Datei/Bibliothekselement/JV im Linemodus des EDT in Ebene 0 bearbeiten. Rückkehr in die CFS-Maske durch @H [ALT] oder @RET.

EDT[*n*]N Wie EDT, jedoch wird das Datenobjekt neu in die angegebene EDT-Ebene eingelesen. Bei Nichtangabe der N-Option wird der Dateninhalt nicht neu eingelesen, falls das Datenobjekt aufgrund eines früheren EDF/EDT Action-Codes noch in der entsprechenden EDT-Ebene steht. Falls der Inhalt der EDT-Ebene durch @D explizit gelöscht wurde, wird in jedem Fall neu eingelesen.

EDT[*n*]R Wie EDT, jedoch wird nach Einlesen des Datenobjekts sofort wieder in die CFS-Maske verzweigt. Die R-Option ist nützlich, falls ein Datenobjekt lediglich für eine spätere Bezugnahme in einer EDT-Ebene abgelegt werden soll (um z.B. einzelne Zeilen in eine andere EDT-Ebene zu kopieren). Es ist auch möglich, Elemente verschiedener Bibliotheken in EDT-Ebenen einzulesen, um sie mit dem EDT-Kommando COMP zu vergleichen oder im SPLIT-Screen Modus des EDT zusammen zu betrachten.

EDT[*n*]S Wie EDT, jedoch wird für die Datei ein File-Kommando mit LINK=EDTSAM abgesetzt. Die Datei wird beim Einlesen in den EDT wie eine Datei mit Nicht-Standard Attributen behandelt. Bei einer ISAM-Datei können die Schlüssel als Bestandteil der Daten im EDT bearbeitet werden. Weitere Informationen, siehe unten (Hinweise).

EDT[*n*]E Wie EDT, jedoch wird vor dem Einlesen des markierten Datenobjekts der bisherige Inhalt der EDT-Ebene nicht gelöscht. Damit ist es möglich, den Inhalt von mehreren Datenobjekten im EDT-Speicher hintereinander anzufügen. Mit diesem Action-Code können z.B. die Inhalte mehrerer Jobvariablen in einer EDT-Ebene zusammen dargestellt werden.

UPD[*n*] Datei/Bibliothekselement/Jobvariable aus dem EDT-Bereich *n* zurückschreiben. Der Action-Code ist nach der Rückkehr aus dem EDT bereits im Action-Feld vorgegeben. Falls das Datenobjekt nicht zurückgeschrieben werden soll, ist der Action-Code mit Blanks zu überschreiben. Die Quittung der einzelnen Updates erfolgt durch die Zeichenfolge U---U am rechten Ende der Bildschirmzeile. Für eine Beschreibung der Datenkonsistenzprüfung vor Ausführung von UPD siehe unten (Hinweise).

Ist der AI-Modus eingeschaltet (Automatic Increment of Version), so wird beim Zurückschreiben in eine LMS-Bibliothek (ISAM) eine neue Version gespeichert, deren Versionsnummer gegenüber der bisher letzten Version um 1 erhöht ist.

EPD[*n*] Gleiche Wirkung wie UPD[*n*], jedoch wird nach dem Update wieder in den EDT verzweigt, um das Datenobjekt weiter zu Bearbeiten. Der Action-Code EPD kann somit zur Zwischensicherung des Datenobjekts verwendet werden.

WPD[*n*] Gleiche Wirkung wie UPD[*n*], jedoch wird die Prüfung auf Datenkonsistenz nicht ausgeführt. Das Datenobjekt wird in jedem Fall zurückgeschrieben, auch wenn das Modification Date ungleich dem beim Einlesen des Objekts gemerkten Datum/Uhrzeit ist. Der Action-Code WPD kann angegeben werden, wenn der Benutzer die Datei im EDT selbst mit dem Kommando Write zurückgeschrieben hat und er danach trotzdem noch den Action-Code UPD ausführen möchte.

Hinweise:

Beim Einlesen einer Datei bzw. eines PLAM-Bibliothekselements in den EDT wird von CFS das Datum und die Uhrzeit der letzten Änderung des Datenobjekts gesichert. Vor Ausführung des Action-Codes UPD wird geprüft, ob das Datenobjekt seit dem Einlesen in den EDT-Datenbereich von einem anderen Benutzer schon geändert wurde. Ist dies der Fall, so wird der Update mit einer entsprechenden Fehlermeldung zurückgewiesen, da sonst die Änderungen des anderen Benutzers überschrieben würden.
Beim Bearbeiten von Dateien mittels Action-Code EDT ist zu beachten, daß das Zurückschreiben dar Datei ausschließlich mit dem Action-Code UPD erfolgen sollte. Bei Verwendung des EDT-Kommandos Write meint CFS, ein anderer Benutzer hat die Datei modifiziert. In diesem Fall kann der Action-Code WPD anstelle von UPD angegeben werden.

Zum Vergleichen der eigenen Änderungen mit den Änderungen des anderen Benutzers wird folgendes Vorgehen empfohlen:

- Der aktuelle Inhalt des Datenobjekts (die Änderungen des anderen Benutzers) ist mit dem Action-Code EDTN in eine andere EDT-Ebene einzulesen (z.B. `EDT1N`).

- Mit EDT-Compare die eigenen Änderungen mit den Änderungen des anderen Benutzers vergleichen (z.B. `COMP0;9` , falls die eigenen Änderungen in der EDT-Ebene 0 vorgenommen wurden).

- Die EDT-Ebene 1 in unserem Beispiel kann, nachdem die eigenen Änderungen mit denen des anderen Benutzers konsolidiert wurden, mit dem Action-Code `UPD1` in das Datenobjekt zurückgeschrieben werden.

Die Prüfung auf Datenkonsistenz vor Ausführung des Action-Codes UPD kann über eine Einstellung im Initialisierungsmodul CFSMAIN ausgeschaltet werden.

Bei unvorhergesehenem Leitungsverlust (allgemein: bei Eintritt des STXIT-Ereignisses ABEND) werden von CFS alle belegten **EDT-Ebenen in Sicherungsdateien gerettet**. Die Sicherungsdateien erhalten die Namen: CFS.EDTSAV [*.jobname*]. *yymmdd.hhmm.x*. Die Ziffer x bezeichnet die Nummer der gesicherten EDT-Ebene. Falls beim LOGON ein Name angegeben wurde, so erscheint dieser ebenfalls in den Namen der erzeugten EDTSAVE-Dateien.
Wenn der Benutzer das nächste Mal nach einem Logon unter der gleichen User-Id und mit dem gleichen Jobnamen CFS aufruft, erhält er einen Hinweis auf das Vorhandensein von EDTSAVE-Dateien.

Mit dem Kommando EDTINIT (siehe Seite **233**) kann nach dem Einlesen eines Datenobjekts die Ausführung vorgegebener EDT-Kommandos veranlaßt werden. Damit ist es möglich, bestimmte **EDT-Voreinstellungen** individuell zu verändern.

Einmal eingestellte EDT-Parameter (z.B. EDIT LONG, SCALE) bleiben über die gesamte CFS-Sitzung erhalten. Das gleiche gilt für die Inhalte der einzelnen EDT-Ebenen, solange keine neuen Dateien in diese Ebenen eingelesen werden. Beim Einlesen einer neuen Datei wird die Zeilen- und Spaltenposition des EDT-Fensters auf den Anfang der Datei gesetzt.

LOWER-Modus des EDT: Beim Einlesen einer neuen Datei/eines neuen Bibliothekselements wird ein zuvor eingegebenes LOWER ON-Kommando zurückgesetzt (LOWER OFF), falls der CAP-Modus in CFS wirksam ist (Standard). Bei eingeschaltetem LOW-Modus von CFS wird der LOWER-Modus des EDT nicht verändert. Ein bei der letzten Dateibearbeitung eingegebenes LOWER ON-Kommando bleibt damit auch für die aktuelle Datei wirksam. Für weitere Informationen siehe Seite **478**.

Werden mehrere Datenobjekte mit EDT[*n*] in verschiedene Ebenen des EDT eingelesen, so wird dem Benutzer die zuletzt angesprochene EDT-Ebene angeboten.

Werden mehrere Datenobjekte mit dem Action-Code EDT markiert, so wird das erste in die edt-Ebene 0, das zweite in die EDT-Ebene 1, usw. eingelesen.

Falls Datenobjekte mit dem Action-Code EDT bearbeitet werden, so setzt CFS im EDT @FILE-Kommandos mit den Namen der eingelesenen Datenobjekte ab. Der Benutzer kann, unter Umgehung des von CFS angebotenen Action-Codes UPD, die bearbeiteten Datenobjekte mit der EDT-Anweisung W[RITE] bzw. SA[VE] selbst wegschreiben.

Non-Standard Dateien (z.B. RECFORM=F-Dateien, Dateien mit BLKSIZE > (STD,1) und ISAM-Dateien mit KEYPOS/KEYLEN ungleich 5/8) können mit EDT bearbeitet werden. Eine manuelle Zuweisung über die Linknamen EDTSAM/EDTISAM ist nicht notwendig, da diese von CFS selbst vorgenommen wird. Die Satzlänge der zu bearbeitenden Daten ist jedoch auf 256 beschränkt.

Mit dem Action-Code EDT[*n*] können mehrere Non-Standard Dateien in verschiedene EDT-Ebenen eingelesen werden. Mit dem Action-Code UPD[*n*] können diese Dateien auch wieder zurückgeschrieben werden.

Mit den Action-Codes EDT/UPD können auch ISAM-Dateien mit **Keylen > 8** und **beliebiger Keypos** editiert werden. Der Isam-Schlüssel wird dabei nicht als Zeilennummer im EDT verwendet, sondern als Teil des Datenbereich eines jeden Satzes angezeigt (LINK=EDTSAM). Die Isam-Schlüssel von Nicht-Standard EDT-Dateien sind damit auch editierbar. Der Benutzer hat in diesem Fall selbst dafür Sorge zu tragen, daß die aufsteigende Reihenfolge der Schlüssel erhalten bleibt.

Durch Betätigung der K1-Taste im Formatmodus wird die Kontrolle wieder an CFS übergeben. K1 beendet den EDT ohne Berücksichtigung von evtl. im gleichen Bildschirm vorgenommenen Datenänderungen.

Durch unbeabsichtigtes Betätigen der K1-Taste während der Dateibearbeitung im EDT gelangt der Benutzer wieder in die CFS-Maske, wobei bei den zuvor markierten Dateien die Action-Codes UPD*n* eingetragen sind. Eine Möglichkeit der Rückkehr an die unterbrochene Stelle im EDT bietet in diesem Fall die F1-Taste. Die Funktion der F1-Taste steht nur solange zur Verfügung, als keine neue Dateienliste selektiert wurde (Kommando NP).

Nach der erneuten Rückkehr aus EDT - nachdem zuvor mit der F1-Taste in den EDT verzweigt wurde - wird der Action-Code UPD von CFS nicht vorbelegt. Zum Zurückschreiben muß der Benutzer den Action-Code UPD ggf. selbst eintragen.

Nach der Rückkehr aus dem EDT muß der Action-Code UPD nicht sofort ausgeführt werden. Bevor UPD ausgeführt wird (evtl. auch auf ein anderes Datenobjekt, können beliebige Anweisungen ausgeführt werden (auch Wechseln der Liste mit NP..), solange die gleiche EDT-Ebene nicht neu gefüllt wird. Die EDT-Ebenen können somit als temporärer Datenspeicher innerhalb einer CFS-Sitzung verwendet werden.

Im Formatmodus des EDT gelangt der Benutzer entweder durch das Kommando HALT oder durch Betätigen der K1-Taste wieder in das Programm CFS zurück. Im HALT-Kommando des EDT kann ein Zusatztext angeben werden: **H [ALT]** *cmd*. Dieser Zusatztext *cmd* wird von CFS als Kommando interpretiert.
Beispiel: H CC1 Es wird aus dem EDT direkt in die Connection 1 verzweigt.

Ein schneller Wechsel zwischen EDT und Connections ist auf folgende Art und Weise möglich: Durch das EDT-Kommando 'H *n*' gelangt der Benutzer aus dem EDT unmittelbar in die Connection mit der Nummer *n*. Der direkte Pfad von einer Connection in die zuletzt mit H *n* verlassene EDT-Ebene erfolgt durch Eingabe der Zeichenfolge

^F1. ^F1 - eingegeben in einer Connection - wirkt so, als ob im Grundprozeß von CFS die F1-Taste gedrückt worden wäre (Bedeutung der F1-Taste: siehe oben).

Bei Verlassen des EDT durch das Kommando H [ALT] *cmd* wird der Update für das Datenobjekt nicht durchgeführt. Der Benutzer muß den Update durch Eintragen des Action-Codes UPD*n* bei dem entsprechenden Datenobjekt nachträglich vornehmen, falls er dies wünscht.

Der Action-Code EDT und die Kommandos EDT/EDL rufen zwei in ihren Datenbereichen vollkommen getrennte EDT's auf. Der Datenbereich des über das Kommando aufgerufenen EDT kann ebenfalls in der oben geschilderten Form als eigener temporärer Datenspeicher verwendet werden.

Unter CFS ist der INTR-STXIT des EDT nicht aktiviert, d.h. das BS2000-Kommando INTR wirkt genauso wie das BS2000-Kommando /R [esume]. Es besteht hierin ein Unterschied zum Aufruf des EDT per BS2000-Kommando: /EXEC EDT.

Falls die EDT-Bearbeitung durch Ausgabe der Job-Report Maske unterbrochen wird, kann nach Beendigung des Job-Report (K1-Taste, END, N) der ursprüngliche EDT-Bildschirm mit der K3-Taste (EDT-Refresh) wieder hergestellt werden.

Wird bei einem Datenobjekt der Action-Code EDT eingetragen, so wird intern zuerst der EDT-Speicher der entsprechenden Ebene gelöscht (@DELETE). Danach wird die Datei eingelesen bzw. das Bibl.-Element satzweise in den EDT-Speicher übertragen. Wird nach der Rückkehr aus dem EDT bei derselben Datei/Bibl.-Element noch einmal der Action-Code EDT eingetragen, so wird der alte, vom Benutzer evtl. schon geänderte EDT-Speicherinhalt weiterhin als gültig angesehen (kein @DELETE und erneutes Laden der Datei). Trifft CFS bei der Action EDT auf einen leeren EDT-Speicher bzw. wurde der Action-Code in der Form EDTN (N=New) angegeben, so wird das Datenobjekt in jedem Fall neu eingelesen.

Ist in CFS der QR-Modus eingeschaltet (Standard), so wird nach erfolgtem Update der virtuelle EDT-Speicher gelöscht. CFS hat keine Kenntnis davon, ob der Update erfolgreich oder nicht erfolgreich ausgeführt wurde. Deshalb wird der virtuelle EDT-Speicher auch bei nicht oder nicht vollständig ausgeführtem Update gelöscht. Durch das Kommando NQR kann das automatische Löschen des virtuellen EDT-Speichers nach UPD unterbunden werden.

Mit den Action-Codes EDT/UPD können **Elemente jedes beliebigen Typs** in PLAM-Bibliotheken bearbeitet werden, insbesondere auch Elemente des Typs R und des Typs C (gebundene Programmphasen). C- und R-Typ Elemente werden von CFS im EDT in einem besonderen Format dargestellt.

Im EDT bearbeitbare Elemente von Programmbibliotheken enthalten normalerweise nur Sätze der Satzart 1. Sollte ein Element jedoch Sätze einer anderen Satzart enthalten, wie z.B Objektmodule (Elementtyp R), so wird diesen Sätzen im EDT der Vorspann "$RT=*xxx*:" vorangestellt. *xxx* ist dabei die Satzart (Record-Type) des betreffenden Satzes. Dabei gilt: 1 <= *xxx* <= 255.

Mit dem Action-Code EDT können auch Elemente aus Programmbibliotheken bearbeitet werden, die Format-A Sätze mit einer Länge > 255 Bytes und/oder Format-B Sätze enthalten. Diese normalerweise nicht im Editor bearbeitbaren Datensätze mit einer Länge größer 256 Bytes werden vor der Übergabe an den EDT in Teilsätze aufgespalten. Folgesätze haben den Vorspann "$RT=CNT:". Format-B Sätze werden vor der Übergabe an den EDT ebenfalls in Teilsätze gespalten. Der erste EDT-Satz für einen Format-B Satz hat den Vorspann "$KEY=*xxxxxxxx*:". *xxxxx* ist dabei die Record-ID des Format-B Satzes. Der zweite und alle folgenden EDT-Sätze für einen Format-B Satz haben den Vorspann "$KEY=C*xxxxxxx*:". *xxxxx* ist die relative Spaltenangabe bezogen auf den Satzanfang. Beim Ändern der Daten ist zu beachten, daß die Gesamtlänge der Daten für einen Format-B Satz ein Vielfaches von 2048 Byte betragen muß.

ESR Es wird eine Maske angezeigt, in der die Eingabedaten für eine Datei-Auswertung bzw. Umsetzung mittels des Dienstprogramms ESR eingetragen werden können. Die angekreuzte Datei wird dabei als Eingabedatei verwendet. Für das Format der anzugebenden ESR-Anweisungen wird auf das ESR-Handbuch verwiesen.

Der Action-Code setzt voraus, daß die ESR-Nachladebibliothek SYSLNK.ESR unter der Kennung TSOS mehrbenutzbar zur Verfügung steht. Falls dies nicht der Fall ist, kann am in der CFSLIB am Beginn des Moduls ESRLES der aktuelle Name der ESR-Nachladebibliothek eingetragen werden.

Systeminformationen zu Datei/Jobvariable/PLAM-Element/Archive-Directory Eintrag anzeigen

F Bei einer Datei wird das Kommando FSTAT *datei*,ALL ausgeführt.
Bei einer Jobvariablen wird das Kommando STAJV *jobvar*,ALL ausgeführt.
Bei einem PLAM-Element oder einem Element in einem ZIP-Archiv wird ein CFS-eigenes Status-Kommando ausgeführt.
Bei einem Archive-Directory Eintrag wird ein CFS-eigenes Status-Kommando ausgeführt.

FP Bei einer Datei wird das Kommando FSTAT *datei*,P ausgeführt. Damit werden alle vorhandenen Dateipassworte ausgegeben. Das gleiche gilt auch für Jobvariablen. Der Action-Code FP ist dem Systemverwalter unter TSOS vorbehalten.

Bei einem Archive-Directory Eintrag werden aufgrund des Action-Codes F folgende Informationen ausgegeben:

```
%  STATUS FOR     :CAT2:$TSOS.LMS.PROC
%
%  SAVE VERSION   =  S.941221.040250
%  TYPE           =  FULL
%
%  FILE VERSION   =  119
%  CFID           =  500A7AFE
%
%  VSN(S)         =  S04451
%  DEVICE         =  TAPE-C4
%
%  EXP-DATE       =  1996-02-01
%
  PLEASE ACKNOWLEDGE
```

Bei einem Element einer PLAM-Bibliothek (z.B. C/CFST) werden aufgrund des Action-Codes F, Informationen in der folgenden Art ausgegeben:

```
%  STATUS FOR     C/CFST
%  NAME        =  CFST
%  TYPE        =  C
%  VERSION     =
%  VARIANT     =  0001
%  CREATION    =  10.05.89  09:35       TSN  5681
%  LAST UPD    =  17.02.90  17:49       TSN  0525
%  LAST ACCESS =  04.08.94  10:31       TSN
%  USER-DATE   =  17.02.90  17:49
%  ELEMENT   IN   HOLD BY USER-ID: CFSTEST
%
%  PROTECTION:    OWNER  = R W X  GROUP  = - - X  OTHERS  = - - X
%                 RDPASS = NO     WRPASS = YES    EXPASS  = NO
%
%  SPACE       =        2  PAM-BLOCKS FOR        1 FORMAT-A RECORD
%                     269  PAM-BLOCKS FOR       35 FORMAT-B RECORDS
%
%  FCBTYPE = PAM ,  RECFORM =  V
%
%  TYPE (TOTAL) OF FORMAT-A RECORDS
%    11 (00001)
%
%  ELEMENT CONTAINS NO SECONDARY NAMES
%  PLEASE ACKNOWLEDGE
```

Datei / PLAM-Element mit openFT übertragen

FT File-Transfer. Datei/PLAM-Bibliothekselement mit openFT in fremdes System übertragen. Falls die für den File-Transfer notwendigen Angaben (Host-Name, fremde Benutzerkennung, usw.) dem FT-System noch nicht bekannt gemacht wurden - entweder durch ein früheres FT-Kommando/Variable Action oder durch einen früheren Action-Code FT -, werden diese Angaben beim ersten FT Action-Code in einer Menue-Maske angefordert. Für eine Beschreibung der Menue-Maske siehe Variable Action ONXFT, Seite 143.

FTM wie Action-Code FT, jedoch wird die Menue-Maske ausgegeben, in der die Angaben für das Zielsystem (z.B. Hostname, User-Id) modifiziert werden können. Die Parameter werden auch für nachfolgende Action-Codes FT verwendet.

FTR wie Action-Code FT, jedoch wird stets die Menue-Maske ausgegeben. In dieser kann neben Hostnamen und User-Id auch der Datei- bzw. Bibliotheks- und Elementname des zu übertragenden Objekts im Zielsystem modifiziert werden.

FTS File-Transfer Status. Es wird ein Kommando NSTAT ... ,INFO=*ALL für die mit dem Action-Code FTS markierte Datei ausgeführt.
Anstelle des Action-Codes FTS kann in einer mit der NSTAT-Option selektierten Dateienliste auch der Action-Code U verwendet werden, um über den momentanen Zustand des FT-Auftrags zu informieren. Bei dem Kommando INSRT U werden die eingefügten Action-Codes U nicht, wie sonst üblich, nach der Ausführung gelöscht.

FTA File-Transfer Activate partner. Es wird ein Kommando MODIFY-FT-PARTNER ..., STATE=*ACTIV für den mit dem Action-Code FTA markierten Partnerrechner ausgeführt. Der Action-Code kann nur unter TSOS und in einer mit NSTAT selektierten Dateienliste eingegeben werden.

FTC File-Transfer Cancel. Es wird ein Kommando NCAN ... für die mit dem Action-Code FTC markierte Datei, bzw. für die angezeigte Transfer-Id ausgeführt. Der entsprechende Transferauftrag wird hiermit annulliert. Der Action-Code kann nur in einer mit NSTAT selektierten Dateienliste eingegeben werden.

FTCF Der Action Code FTCF setzt zwei CANCEL-FILE-TRANSFER Kommandos für die entsprechende Transfer-ID ab: Das erste mit dem Parameter FORCE-CANCEL=*NO und das zweite mit dem Parameter FORCE-CANCEL=*YES.

FTQF File Transfer Queue First. Der markierte File Transfer Auftrag als nächster ausgeführt.

FTQL File Transfer Queue Last. Der markierte File Transfer Auftrag als letzter ausgeführt.

Die Action-Codes FTQF/FTQL sind erst ab openFT V6.0 zulässig. Voraussetzung ist außerdem, daß die Dateienliste mit der NSTAT [*ALL] im Feld FILENAME-SELECT erzeugt wurde.

FTHP File Transfer High Priority. Der markierte File Transfer Auftrag wird bezüglich anderer anstehender Aufträge als hochprior behandelt.

FTLP File Transfer Low Priority. Der markierte File Transfer Auftrag wird bezüglich anderer anstehender Aufträge als niederprior behandelt.

FTNP File Transfer Normal Priority. Der markierte File Transfer Auftrag wird bezüglich anderer anstehender Aufträge als normalprior behandelt.

Die Action-Codes FTHP/FTLP/FTNP sind erst ab openFT V6.0 zulässig. Voraussetzung ist außerdem, daß die Dateienliste mit der NSTAT [*ALL] im Feld FILENAME-SELECT erzeugt wurde.

Hinweise:

Zur Selektion aller angestarteten und noch nicht beendeten FT-Aufträge wird auf die NSTAT-Option im Feld FILENAME-SELECT der Selektionsmaske (Seite 56) hingewiesen.

Bei der Übertragung von Dateien mit dem Action-Code FT werden Read-Passworte für die zu übertragenden Dateien von CFS dynamisch nachgefordert.

Die Beendigung von FT-Aufträgen kann mit dem Job-Report System von CFS überwacht werden (FT-Parameter Protocol-Listing: YES). Für weitere Informationen siehe Kommando JRFT/NJRFT auf Seite 412.

Für weitere Hinweise zum File-Transfer, insbesondere File-Transfer zwischen BS2000 und SINIX, siehe Kapitel 12 "File-Transfer mit openFT/RDAC".

PLAM-Bibliothekselement ausleihen/Ausleihung zurücksetzen

HOLD | NHOLD

Ab LMS Version 3.0 können Elemente einer Bibliothek ausgeliehen werden. Dies bedeutet, daß die Benutzerkennung des Entleihers im Inhaltsverzeichnis der Bibliothek vermerkt wird. Nur Benutzer, die unter der Kennung des Entleihers angeloggt sind, können das Element in der gleichen oder in einer höheren Version ändern oder kopieren.

Mit dem Action-Code HOLD wird ein Element ausgeliehen. Hierbei ist es von Bedeutung, unter welcher Benutzerkennung die Ausleihung erfolgt.

Mit dem Action-Code NHOLD wird der Ausleihstatus für ein Element zurückgesetzt.

Hinweis:

Für eine ausführliche Beschreibung der Auswirkungen von HOLD wird auf das LMS-Benutzerhandbuch Version 3.0 verwiesen: Kapitel 3 (Elemente), Abschnitt 3.6 (Elementschutz/Datenschutz), Thema: "Übersicht über notwendige Rechte für LMS-Aktionen".

Dateibeschreibung im INF-Katalog anzeigen / ändern

I Inform. Anzeigen der zu dem Datenobjekt (Datei/Jobvariable/Bibliothekselement) eingegebenen Dokumentation.

IM [*f*] Information Modify. Neuerfassung/Update eines Dokumentationstextes.

f Nur bei Neuerfassung eines Dokumentationstextes: 1- bis 3-stelliger mnemotechnischer Name eines Formats für die Erfassung des Dokumentationstextes.
Die Maskenformate für die Erfassung von Dokumentationstexten und die Mnemo-Codes, über die diese Masken angesprochen werden, sind vom Systemverwalter in einem zentralen Katalog CFS.INF.FORMAT einzutragen.

IA [*f*] Information Add. Anfügen einer zusätzlichen Seite von Dokumentationstext.

f 1- bis 3-stelliger Name eines Formats für die Erfassung der Dokumentation.

Hinweis: Falls für das betreffende Datenobjekt noch keine Dokumentation gespeichert ist, so wirkt der Action-Code IA wie IM (Neuerfassung).

II [*f*] Information Insert. Einfügen einer zusätzlichen Seite von Dokumentationstext. Die neue Seite an Informationstext wird hinter der letzten, mit dem gleichen Format erfaßten Informationsseite eingefügt.

f 1- bis 3-stelliger Name eines Formats für die Erfassung der Dokumentation.

Hinweis: Falls für das betreffende Datenobjekt noch keine Dokumentation gespeichert ist, so wirkt der Action-Code II wie IM (Neuerfassung).
Falls das im II Action-Code angegebene Format im bestehenden Informationstext nicht vorkommt, so wird die neue Textseite am Ende der Beschreibung eingefügt (gleiche Wirkung wie IA: Information Add).

IE Information Erase. Löschen der zu dem Datenobjekt gespeicherten Dokumentationen.

IEQ Information Erase with Query. Die einzelnen Seiten des Informationstextes werden wie beim Action-Code I (Inform) angezeigt. Die angezeigte Informationsseite wird durch Drücken der ENTER-Taste gelöscht. Die Löschung der angezeigten Informationsseite wird durch Betätigung der K3-Taste unterbunden. In beiden Fällen (ENTER/K3) wird dem Benutzer die nächste Informationsseite angeboten. Durch Drücken der K1-Taste kann der Vorgang abgebrochen werden. Die zuletzt angezeigte Informationsseite wird in diesem Fall nicht gelöscht.

IC Information Copy. Kopieren des gesamten Informationstextes. Es wird die Eingabe des Namens eines bereits dokumentierten Datenobjekts verlangt. Die von diesem Datenobjekt kopierte Dokumentation kann anschließend modifiziert werden (implizites IM). Für nähere Informationen siehe Seite 421 und folgende "CFS Dateiinformationssystem".

Datei/Bibliothekselement/Jobvariable im Änderungsmodus anzeigen

M Ändern des Inhalts eines Datenobjekts (Datei/Jobvariable/Bibliothekselement).

Über den Action-Code M wird das CFS-Display/Editor-System aufgerufen und die Datei/das Bibliothekselement im Änderungsmodus eröffnet. Der Modify-Modus bleibt solange wirksam, bis er durch das Kommando NM (No Modify) ausgeschaltet wird, bis durch das Kommandos NF (Next File) die nächste mit D (Display) oder M (Modify) markierte Datei angezeigt wird, bzw. bis das Editor-System durch Drücken der K1-Taste ganz verlassen wird.

Falls die Datei/das Bibliothekselement im Lesemodus eröffnet werden soll, so ist der Action-Code D (Display) anzugeben.

Für eine ausführliche Beschreibung des CFS-Display und Editor-Systems wird auf Seite **289** und folgende "CFS-Display/Editor" verwiesen.

Hinweise:

Beim Modify für sehr große Bibliothekselemente kann es zu Platzproblemen im PUBLIC-Speicher kommen, da CFS für das Element eine temporäre Datei mit dem Namen #Z[*.typ*] [*.elementname*] anlegt.

Modify-Actions werden ebenso wie Display-Actions in der Reihenfolge der Eingabe sofort ausgeführt. Um von der Anzeige einer Datei zur Anzeige der nächsten Datei überzugehen, ist im Kommandofeld von CFS der Befehl NF (Next File) bzw. D einzugeben. Mit dem Befehl LF (Last File) kann wieder zur Anzeige der vorhergehenden Datei zurückverzweigt werden. Möchte der Benutzer aus dem Modify-Modus wieder in die CFS-Dateienliste zurückkehren, so ist das Kommando LST (Liste) anzugeben bzw. die K1-/F1-Taste auszulösen.

Administrationsrechte für eine PLAM-Bibliothek anzeigen / ändern

Vorbemerkungen:

Die i.f. beschriebene Möglichkeit der Vergabe von Administrationsrechten für eine PLAM-Bibliothek, setzt eine LMS-Version >= 2.0 und BS2000 >= 11.0 voraus.

Der Action-Code LA [*xx*] kann in der Dateienliste bei PLAM-Bibliotheken (FCBTYP = PL) und bei beliebigen Elementen einer PLAM-Bibliothek angegeben werden. Die Wirkung ist in jedem Fall die gleiche: Es werden die Administrationsrechte der aktuellen Bibliothek angezeigt bzw. verändert. Befindet sich der Benutzer innerhalb einer Bibliothek, so ist das Element, bei dem der Action-Code angegeben wird, ohne Bedeutung.

LA

Es werden die Administrationsrechte, die Standard Elementschutzattribute sowie die Standard Schutzattribute für Elementtypen für die aktuelle Bibliothek angezeigt. Die Informationen werden in folgendem Format dargestellt:

```
%  SHOW LIBRARY ADMINISTRATION + STANDARD ELEMENT-PROTECTION
%
%  LIBRARY NAME            : CFSLIB
%
%  LAST ACCESS DATE        : YES              (Modify with:  LAKL/LANKL)
%
%  LIBRARY ADMINISTRATION : OWNER   = YES     GROUP   = NO      OTHERS   = NO
%                            PASSW   = NO
%
%  STD ELEMENT-PROTECTION : OWNER   = R W *   GROUP   = - - *   OTHERS   = - - *
%                            RDPASS = NO      WRPASS = NO       EXPASS   = NO
%
%  STD PROTECTION TYPE C/ : OWNER   = R W X   GROUP   = - - X   OTHERS   = - - X
%                            RDPASS = NO      WRPASS = NO       EXPASS   = NO
%                            ADMIN   = OWNER           OTHERS
%                            WRITE-CONTROL = YES
%
%  PLEASE ACKNOWLEDGE
```

Hinweise:

Das Zeichen '*' in der Protection-Zeile zeigt an, daß für die entsprechende Zugriffsart keine Basic Access Control Liste definiert ist. Dies bedeutet, daß Owner, Group und Others Zugriff in der entsprechenden Art auf das Element haben.

Die Definition von Standard Element- und Elementtyp-Schutzattributen ist mit dem Action-Code SP möglich. Näheres hierzu auf Seite **204**.

LA *x* [*t*/]

Administration der gesamten Bibliothek bzw. der Elemente eines Typs bestimmten Benutzern erlauben. Das Administrationsrecht gestattet es, Elemente neu aufzunehmen, zu löschen, zu kopieren oder umzubenennen. Das Administrationsrecht kann dem Eigentümer (USER, d.h. der Benutzerkennung, unter der die Bibliothek katalogisiert ist), der Benutzergruppe, der der Eigentümer angehört (GROUP) oder allen anderen Benutzerkennungen (OTHERS) zugeteilt werden.

x

Domäne, der die Administration der Bibliothek erlaubt werden soll.

- U User. Der Eigentümer (alle Benutzer der Kennung, unter der die Bibliothek katalogisiert ist) hat das Recht zur Administration.
- G Group. Eine festgelegte Menge von Benutzerkennungen (Benutzergruppe) soll das Administrationsrecht besitzen.
- O Others. Alle Kennungen, die nicht Eigentümer der Bibliothek sind oder der Benutzergruppe des Eigentümers angehören, sollen das Administrationsrecht besitzen.

Für *x* kann eine beliebige Kombination der Buchstaben U, G, O angegeben werden. Z.B. bewirkt der Action-Code LAUG, daß die Benutzerkennung, unter der die Bibliothek katalogisiert ist und die Benutzergruppe das Administrationsrecht für die Bibliothek besitzen.

t/

Die zuvor angegebene Domäne (U/G/O) soll das Administrationsrecht für Elemente des Typs *t* erhalten. Nur Benutzer dieser Domäne haben das Recht, Elemente des Typs *t* zu ändern, zu löschen oder umzubenennen.
Durch wiederholtes Eintragen des Action-Codes kann das Administrationsrecht mehreren Domänen zugeteilt werden. Beispiel: LAUS/ LAOS/ erlaubt dem Owner (User) und der Domäne Others die Administration der Elemente des Typs S.
Das typbezogene Administrationsrecht für Elemente steht erst ab LMS Version 3.0 zur Verfügung.

NLA *t*/

Es wird eine zuvor definierte Administrationsberechtigung für Elemente des Typs *t* zurückgesetzt. Für diesen Typ gelten dann die globalen Administrationsrechte. Anstelle von NLA *t*/ kann auch LAN *t*/ angegeben werden.

Der Action-Code LA kann bei einem beliebigen Element der Bibliothek eingetragen werden. Die Wirkung ist in allen Fällen die gleiche.

Im Standardfall, d.h. ohne besondere Definition eines Administrationsrechts für einen Elementtyp gelten die globalen Administrationsrechte der Bibliothek.

NLA *x*

Recht der Administration der Bibliothek für Benutzerkennungen verbieten.
Ohne Administrationsrecht kann der Benutzer Elemente nicht neu aufnehmen, löschen, kopieren oder umbenennen. Anstelle von NLA *x* kann auch LAN *x* angegeben werden.

x Domäne, der die Administration der Bibliothek nicht erlaubt werden soll. Beschreibung siehe oben. Für *x* kann auch eine Kombination der Buchstaben U, G, O angegeben werden.

NLA Administrationsrechte für die Bibliothek werden zurückgesetzt. Jeder Benutzer, der gemäß dem DVS Zugriff auf die Bibliothek hat, kann Elemente aufnehmen, löschen, kopieren oder umbenennen. Anstelle von NLA kann auch LAN angegeben werden.

LAP [*t*/] Es wird eine Maske angefordert, in der der Benutzer ein bis zu vier Byte langes Administrations-Passwort für die gesamte Bibliothek bzw. für alle Elemente eines Typs /*t* vergeben kann. Administrationstätigkeiten in der Bibliothek (siehe oben) sind dann nur möglich, falls zuvor durch ein Password-Kommando das benötigte Password abgesetzt wurde. Die Möglichkeit der Vergabe eines Administrationspasswortes ist unabhängig von der Definition bestimmter Administrationsdomänen mit dem Action-Code LA *x* (siehe oben).
Die Eingabe des Administrations-Passworts erfolgt in der Form: [C]'...' oder X'...'.

Da Passworte in PLAM-Bibliotheken in verschlüsselter Form gespeichert werden, kann der Wert eines Passworts im Nachhinein weder vom Systemverwalter, noch vom Eigentümer der Bibliothek ermittelt oder abgefragt werden. Für ein nicht mehr bekanntes Passwort bleibt für den Eigentümer der Bibliothek nur die Möglichkeit, dieses zu entfernen und evtl. neu zu vergeben.

NLAP Der Eigentümer, d.h. ein Benutzer, der unter der User-ID angeloggt ist, unter der die Bibliothek katalogisiert ist, kann ein vorhandenes Administrations-Passwort entfernen. Es ist hierzu nicht die Kenntnis des betreffenden Passworts notwendig. Anstelle von NLAP kann auch LAPN angegeben werden.

LAKL | NLAKL

Keep Last Access. Das Aktualisieren des Last Access Date für Zugriffe auf Elemente der Bibliothek wird ein-/ausgeschaltet.
Die Variante KL des Action-Codes LA ist ab LMS Version 3.0 zulässig.
Im Modul CFSMAIN kann der Systemverwalter eine Option aktivieren, die bewirkt, daß von CFS neu angelegte Bibliotheken automatisch mit Last Access Date eingerichtet werden.

LAWC | NLAWC

Die Eigenschaft Write-Control für alle Elemente mit dem gleichen Typ wie das mit dem Action-Code LAWC markierte Element ein-/ausschalten.
Die Variante WC des Action-Codes LA ist ab LMS Version 3.0 zulässig.

Bei eingeschaltetem Write-Control für einen Elementtyp *t* gelten folgende Zugriffsbeschränkungen für Elemente des Typs *t*:

- Falls ein Element *t/element/version* nicht ausgeliehen, d.h. im Zustand HOLD ist, ist es nicht möglich, dieses Element zu ändern oder zu überschreiben.
- Unabhängig vom Hold-Status ist es nicht möglich, ein Element *t/element/ version* umzubenennen.
- Das Element *t/element/version* darf nicht ausgeliehen sein (Hold-Status), damit aus einer Version eine höhere Version erzeugt werden kann.

Hinweis:

Falls mit dem Action-Code LA.. das Administrationsrecht für die PLAM-Bibliothek verändert wurde, werden von CFS für diese Bibliothek auf Dateiebene die folgenden zusätzlichen Prüfungen durchgeführt: Besitzt die Datei das Attribut SHARE=YES ?. Wenn Ja: Ist für die Datei eine Basic Access Control Liste (BACL) oder eine Full Access Control Liste (FACL) definiert ? Im negativen Falle werden für die Datei die BACL-Attribute OWNER/GROUP/OTHERS = READ/WRITE/EXEC gesetzt. Damit ist sichergestellt, daß die Datei nicht mehr kopiert werden kann.

Guard-Schutz für Dateien / Bibliothekselemente festlegen

MG Modify Guards. Es wird eine Maske ausgegeben, in der die aktuellen Guard-profile der Datei angezeigt und geändert werden können.

Für weitere Hinweise zum Guard-Schutz siehe Action-Code SG.

HSMS: Migration=Allowed/Inhibited

MA | **MI** Es wird das Dateiattribut MIGRATION= für die bezeichnete Datei geändert.

MA MIGRATION=ALLOWED: Die Datei darf durch HSMS ausgelagert werden.

MI MIGRATION=INHIBITED: Die Datei darf nicht ausgelagert (migriert) werden.

Zum Thema HSMS siehe auch Seite 89 (User Option HSMS) bzw. 92 (User Option MIGRATE).

Inhaltsverzeichnis einer Bibliothek/eines Archive-Directories anzeigen
Alle Versionen eines Elements anzeigen
Alle Archive-Sicherungsversionen einer Datei anzeigen

NP Dieser Action-Code kann in der Dateienliste eingegeben werden bei:

- einer Bibliothek: Es wird das Inhaltsverzeichnis dieser Bibliothek als neue Dateienliste angezeigt.

- einem ZIP-Archiv: Es wird das Inhaltsverzeichnis dieses ZIP-Archivs als neue Dateienliste angezeigt.

- einem als Datei angezeigten Archive-Directory: Es werden die jeweils letzten (aktuellsten) Sicherungseinträge für Dateien der eigenen Kennung aus diesem Directory angezeigt. Siehe auch Action-Code NPA weiter unten.

- einem Eintrag in einer Archive-Directory Liste: Es werden alle im Archive-Directory enthaltenen Sicherungseinträge zu der markierten Datei angezeigt. CNS-Einträge ("Catalogued but not Saved") werden hierbei nicht berücksichtigt. Siehe auch Action-Code NPA weiter unten.

- einem in Delta-Form gespeicherten Element einer PLAM-Bibliothek: Es werden alle Versionen des Elements ausgewählt (User Option DELTA).

- einem nicht Delta-gespeicherten Element einer PLAM-Bibliothek. Es werden alle Versionen des Elements angezeigt.

NP [-] *t* [/] Bei PLAM-Bibliothek: Inhaltsverzeichnis nur von Elementen des angegebenen Typs bzw. von allen Elementen außer dem angegebenen Typ anzeigen. Durch NPS/ werden z.B. alle Quellprogramme mit Elementtyp S angezeigt. Durch NP-S/ werden alle Elemente mit Ausnahme von Typ S angezeigt.
Der abschließende Schrägstrich nach der Typangabe kann weggelassen werden, falls ein Typ ungleich F oder L ausgewählt wird.

Die Action-Codes NP/NPL/NPF haben die gleiche Wirkung wie das Kommando NPLMS/LMR/FMS/DIR *bibliothek*.
Der Action-Code NP*t*/ hat die gleiche Wirkung wie NPLMS *bibliothek*;FCB=*t*.
Bei deltagespeicherten PLAM-Elementen hat der Action-Code NP die gleiche Wirkung wie das Kommando NPLMS *bibliothek*,*element*;U=DELTA.

NPA Bei einem als Datei angezeigten Archive-Directory: Falls der Benutzer unter TSOS eingeloggt ist, werden aus diesem Directory die aktuellen Sicherungseinträge für alle Benutzerkennungen angezeigt.

Bei einem Eintrag in einer Archive-Directory Liste: Es werden alle im Archive-Directory enthaltenen Sicherungseinträge zu der markierten Datei angezeigt. CNS-Einträge ("Catalogued but not Saved") werden hierbei ebenfalls berücksichtigt.

Datei / Bibliothekselement ausdrucken

P	Print	
PS	Print mit Parameter	SPACE=E.
PB	Print mit Parameter	BINARY=YES.
PE | PSE	Print mit Parameter	ERASE.
P*n*	*n*-maliges Ausdrucken der Datei/des Bibliothekselements. *n* kann nach jeder Variante der Print-Action angegeben werden.	

Der **Ausdruck von Jobvariablen** kann nur über die Variable Action ONXLIST oder über den Hardcopy-Modus beim Display der Jobvariablen erfolgen.

Alle Tasks anzeigen, die eine Datei eröffnet haben

OPN Es werden alle Tasks (TSN's) angezeigt, die die Datei gegenwärtig eröffnet haben (Verwendung der User Option: `OPEN IOENPRSEC`). Dieser Action-Code setzt voraus, daß der Systemverwalter unter TSOS den CFS-Holdertask gestartet hat (/ENTER CFS.S. LMSLIB(CFSHT)). Falls die Datei im Output-Modus eröffnet ist (SAM-/PAM-Datei), so kann der aktuelle Inhalt mit dem Action-Code D (Display) angezeigt werden. Mit dem Action-Code S kann ein Status-Kommando auf die im Listeintrag angegebene TSN ausgeführt werden.
Falls es sich bei der Datei um eine PLAM-Bibliothek handelt, werden alle Tasks angezeigt, die diese Bibliothek geöffnet haben, sowie die geöffneten Elemente dieser Bibliothek. Der Action-Code D ist auf Elemente nicht möglich.
Bei Jobvariablen werden alle Tasks angezeigt, die ein ONEVT-Ereignis mit dieser Monitor-Jobvariablen offen haben.

Ausdruck auf Stationsdrucker/Laserdrucker, File-Transfer, RDAC

PD*n* Print on Device. Über die vom Systemverwalter einzurichtende Datei CFS.PDFILE wird eine RZ-spezifische Beziehung hergestellt zwischen dem im PD-Action-Code angegebenen ein- bis zweistelligen Mnemo-Code *n* und dem Stationsnamen eines Remote-Druckers. Über einen Mnemo-Code *n* können auch Print-Kommandos mit besonderen Parametern (z.B für Laserdrucker: FORM=, LOOP=, CHARS=, ..) und File-

Transfer/RDAC-Aufrufe zu vorgegebenen Host-Rechnern angefordert werden. Das Satzformat der CFS.PDFILE ist im Handbuch für den Systemverwalter, Abschnitt "Administrationsschnittstellen" dokumentiert.

Durch das Kommando /FILE *datei*,LINK=PDFILE kann sich jeder Benutzer auch eine eigene, auf seine besonderen Bedürfnisse angepaßte PDFILE zuweisen. Das FILE-Kommando kann bereits beim Startup von CFS angegeben werden.

PD*n*S	PD mit Parameter	SPACE=E
PD*n*E	PD mit Parameter	ERASE
PD*n*B	PD mit Parameter	BINARY=YES

PD? Es erscheint ein Menü mit den vom Systemverwalter eingerichteten Mnemo-Codes und den zugeordneten Druckern/Print-Parametern. Der gewünschte Drucker wird mit X bzw. S (Print mit Space=E) markiert.

PD erstmalige Angabe: gleiche Wirkung wie PD?, d.h. es erscheint ein Menü mit den vom Systemverwalter eingerichteten Mnemo-Codes und zugeordneten Druckern/ Print-Parametern. Der Benutzer kann den gewünschten Drucker mit X bzw. S (Print mit Space=E) markieren. Im Unterschied zu PD? merkt sich CFS den angekreuzten Mnemo-Code und setzt diesen bei allen folgenden Action-Codes PD ein.

Datei / Bibliotheks- / ZIP-Element / Jobvariable umbenennen

R Rename. Der Name des durch den Action-Code R markierten Datenobjekts wird in der Maske hell dargestellt und kann vom Benutzer modifiziert werden durch einfaches Überschreiben mit dem gewünschten neuen Inhalt.

Durch Drücken der Taste **F2** werden die Namen aller gerade am Bildschirm angezeigten Datenobjekte überschreibbar und können so verändert werden.

Eine ordnungsgemäße Ausführung der Rename-Operation wird durch die Quittung R---R ganz rechts im entsprechenden Listeintrag angezeigt. Im Fehlerfall wird hier der DVS oder Bibliotheks-Fehlercode eingetragen.

Datei mit RDAC übertragen / auf Remote-Drucker ausgeben

RD Datei mit RDAC/FTOS in fremdes System/zu einem PC übertragen/auf einem Remote-Drucker ausgeben. Mit dem Action-Code RD können auch PLAM-Bibliothekselemente auf einen Drucker ausgegeben werden. Falls die für den File-Transfer notwendigen Angaben (Host-Name, fremde Benutzerkennung, usw.) dem FT-System noch nicht bekannt gemacht wurden - entweder durch ein früheres RDAC-Kommando/Variable Action oder einen Action-Code RD -, so werden die Angaben beim ersten RD Action-Code in einer Menue-Maske angefordert. Für eine Beschreibung der Maske siehe Seite **Fehler! Textmarke nicht definiert.**: Variable Action ONXRDAC.

RDS Bei Ausgabe der Datei auf einem Remote-Drucker: Ausdruck der Datei mit dem Parameter SPACE=E.

RDM RDAC-Transfer mit explizit angeforderter Menue-Maske (siehe oben).

Für weitere Hinweise zum File-Transfer mit RDAC - insbesondere File-Transfer zwischen BS2000 und MS-DOS-Systemen wird auf das Kapitel 12 "File-Transfer mit openFT/RDAC" verwiesen.

Bibliothek / ZIP-Archiv / ISAM-Datei reorganisieren

REO Reorganisieren. Die bezeichnete Bibliothek, ISAM-Datei, bzw. das ZIP-Archiv wird reorganisiert, indem alle Elemente in eine neu angelegte Hilfsdatei übertragen werden. Diese Hilfsdatei wird am Ende auf die ursprüngliche Bibliothek (ISAM-Datei) zurückkopiert und mit /FILE ...,SPACE=-9999 wird nicht mehr benötigter Speicherplatz freigegeben. Die Hilfsdatei wird anschließend gelöscht. Durch die REO-Action werden die durch häufiges Löschen und Neuaufnehmen von Elementen entstandenen "Löcher" aus der Bibliothek entfernt. Die Bibliothek wird "gepackt" und beansprucht am Ende weniger Speicherplatz als vor der Reorganisation. Für weitere Informationen zum Reorganisieren von Dateien siehe Seite 268.

REOL analog zu dem Parameter MODE=LMS im CFS-Kommando REORG.
Das Unterprogramm LMS wird aufgerufen. LMS reorganisiert inplace, d.h. ohne die einelnen Elemente in eine neue Bibliothek zu übertragen und am Ende die Bibliothek zu kopieren. Mit dieser Methode wird eine Bibliothek in kürzestmöglicher Zeit reorganisiert. Unter Umständen werden weniger Seiten freigegeben als bei der CFS-Methode.
Standard: REOC (CFS-eigene Reorganisationsmethode).

Dateikettungsnamen freigeben

REL Release Link. Falls im Feld FILENAME-SELECT mit der Option **RDTFT** alle Dateien ausgewählt wurden, die durch einen Linknamen mit dem aktuellen Prozeß verbunden sind, wird durch den Action-Code REL der in der Spalte LINK aufgeführte Linkname durch ein REL-Kommandos freigegeben.

RETPD für Datei/Jobvariable verändern

RP *n* Retention Period (Schutzfrist) für Datei/Jobvariable setzen.

n 0 <= *n* <= 999: Zeitraum als Anzahl von Tagen, während dessen die Datei/ Jobvariable gegen Veränderungen geschützt ist.

Select: Bibliothekselement in Datei umwandeln

S Select. Aus dem mit S markierten Bibliothekselement wird eine Datei gleichen Namens erzeugt. Alle in der Maske der Dateienliste eingetragenen Action-Codes S werden wie Variable Actions gesammelt und am Ende der Bearbeitung ausgeführt. Siehe hierzu auch Kommando A (Ausführen) auf Seite 215.

Der Name der erzeugten Datei ist gleich mit dem Namen des markierten Bibliothekselement. Die Elementtyp- und Versionsbezeichnungen werden nicht zur Namensbildung herangezogen.

Beim Selektieren eines PLAM-Bibliothekselements in eine Datei werden nur Elementsätze der Satzart 1 berücksichtigt. Sätze mit anderen Satzarten, wie sie z.B. in Elementen vom Typ R vorkommen, werden nicht in die Datei übertragen. Für weitere Informationen zu den im Element gespeicherten Satzarten siehe Action-Code F.

SR Select with Rename. Der Name des Bibliothekselements wird modifizierbar und hell gesteuert und kann mit dem Namen der zu erzeugenden Datei überschrieben werden. Der Action-Code SR wird im Gegensatz zu S sofort ausgeführt.

Guard-Schutz für Dateien / Bibliothekselemente festlegen

SG Set Guards. Datei/PLAM-Bibliothekselement mit Guards schützen. Falls die Namen der Guard-Profile (Read-/Write-/Exec-Guard) CFS noch nicht bekannt gemacht wurden - entweder durch ein früheres GUARD-Kommando/Variable Action oder durch einen früheren Action-Code SG/SGM -, werden die Namen beim ersten SG Action-Code in einer Menue-Maske angefordert. Für eine Beschreibung der Menue-Maske siehe Variable Action ONXGUARD, Seite 147.

SGM wie Action-Code SG, jedoch wird in jedem Fall die Guards-Maske ausgegeben.

SGN Set Guards None. Der Guard-Schutz wird für das betreffende Datenobjekt außer Kraft gesetzt. Bei Dateien kann dies auch durch das BS2000-Kommando /MODIFY-FILE-ATTR *datei*,PROTECTION(GUARD=*NONE) erreicht werden.

MG Modify Guards. Es wird eine Maske ausgegeben, in der die aktuellen Guard-profile der Datei angezeigt und geändert werden können.

Für weitere Hinweise zum Guard-Schutz siehe Seite 247.

Show Index Attributes (NK-ISAM)

SI Show Index Attributes. Es werden die Namen und die Keypos/Keylen aller evtl. bestehenden Sekundärindizes einer NK-ISAM Datei angezeigt.

Zur Behandlung von NK-ISAM Dateien mit Sekundärindizes ist auch die User Option INDEX (Seite 90) und das Kommando DA (Seite 302) von Interesse.

Standard-Schutzattribute für neue Elemente einer PLAM-Bibliothek festlegen

Vorbemerkung:

Die im folgenden beschriebene Möglichkeit der Vergabe von Standard Schutzattributen für neue Elemente einer PLAM-Bibliothek setzt eine LMS-Version >= 2.0 und BS2000 >= 10.0 voraus.

SPSTD Set Protection Standard. Die Schutzattribute (BACL's) des mit SPSTD bezeichneten Bibliothekselements werden als Standard-Schutzattribute für alle neu erzeugten Elemente in der PLAM-Bibliothek verwendet.

Die Standard-Schutzattribute gelten nicht für neue Elemente des Typs *t*/, für diesen Elementtyp eigene Standardschutzattribute definiert wurden (siehe unten, Action-Code SP*t*/). Die Standard-Schutzattribute gelten ebenfalls nicht für Elemente, die mit dem Action-Code CS (Copy Same) kopiert, bzw. mit der Variablen Action ONXADD.. ,CPR (Copy Protection) aus einer anderen Bibliothek übertragen wurden.

Die aktuell gültigen Standard-Schutzattribute werden durch den Action-Code LA (Library Attributes, siehe Seite 197) angezeigt.

SP *t*/ Set Protection for Type *t*. Die Schutzattribute (BACL's) des mit dem Action-Code SP bezeichneten Bibliothekselements werden als Standard-Schutzattribute für alle in der PLAM-Bibliothek neu erzeugten Elemente mit dem Elementtyp *t* verwendet. Sind für das bezeichnete Bibliothekselement Write-, Read-, Exec-Passworte definiert, so werden diese nicht in die Standard-Schutzattribute für den Elementtyp übernommen.

SPN *t*/ — Die Standard-Schutzattribute für neue Elemente des angegebenen Typs werden zurückgesetzt. Die typunabhängigen Standard-Schutzattribute (siehe Action-Code SPSTD) bleiben nach wie vor erhalten.

SPN — Die Standard-Schutzattribute für neue Elemente in der Bibliothek werden zurückgesetzt. Die Standard-Schutzattribute für für bestimmte Elementtypen bleiben dabei nach wie vor bestehen.

Hinweise:

Sind für das Bibliothekselement Passworte definiert, so werden diese bei der Abarbeitung des Action-Codes SP angefordert und wie eingegeben in die Standard Protection übernommen.

Falls mit dem Action-Code SP.. Standard-Schutzattribute für die PLAM-Bibliothek definiert oder verändert wurden, werden von CFS für diese Bibliothek auf Dateiebene die folgenden zusätzlichen Prüfungen durchgeführt: Besitzt die Datei das Attribut SHARE=YES ?. Wenn Ja: Ist für die Datei eine Basic Access Control Liste (BACL) oder eine Full Access Control Liste (FACL) definiert ? Im negativen Falle werden für die Datei die BACL-Attribute OWNER/GROUP/OTHERS = READ/WRITE/ EXEC gesetzt. Damit ist sichergestellt, daß die Datei mit dem Copy-Kommando nicht mehr kopiert werden kann.

SHARE=Yes/No für Dateien / Jobvariablen

SY | SN — Share=Yes / Share=No für die entsprechend markierte Datei/Jobvariable.
SH | NS — SHareable / NonShareable. Synonym zu SY/SN.

SHS — SHare=Special. Die Datei ist auch für die Benutzerkennung Service zugänglich.

SHR — SHare=Yes und Access=Read.

Eintrag in Dateienliste aktualisieren (Update)

U — Update File-List. Für die mit dem Action-Code U markierten Datenobjekte werden in der Dateienliste die Spalten LASTP, TYP, AGE, PASS, usw. aktualisiert.
Bei deltagespeicherten PLAM-Elementen wird wird durch den Action-Code U die höchste momentan gespeicherte Version des Elements angezeigt.

Der Action-Code U kann z.B. im Zusammenspiel mit der User Option NO (Names Only) angewendet werden, falls den Benutzer die in der Dateienliste zunächst leergelassenen Attribute wie Dateigröße, Fcbtyp, Alter für einzelne Dateien doch interessieren. Durch Eintragen des Action-Codes U bei den betreffenden Datenobjekten werden die angesprochenen Eigenschaften in der Dateienliste eingetragen.

Variable Action zur Ausführung vormerken

X — Die in der Form ONX ... definierte Variable Action wird zur späteren Ausführung vorgemerkt. Wurde die Variable Action in der Form ON& ... definiert, so ist der Action-Code X nicht notwendig, da die Variable Action in diesem Fall automatisch für alle Elemente der Dateienliste ausgeführt wird.

XE — Variable Action vormerken und nach erfolgreicher Ausführung das Datenobjekt löschen. Konnte die Variable Action nicht ordnungsgemäß ausgeführt werden, so wird der Zusatz E im Action-Code ignoriert.
Die ERT-Option (Erase with Retain Tempfiles) wird bei XE nicht aktiv, da der Benutzer mit XE die Datei ausdrücklich zum Löschen freigegeben hat.

X*t* Nur bei Aufnahme von Dateien/Bibliothekselementen in eine PLAM-Bibliothek: Die Datei wird unter dem angegebenen Elementtyp in der PLAM-Bibliothek gespeichert. Der Buchstabe E ist als Typ unzulässig, da der Action-Code XE von CFS wie oben beschrieben als Erase interpretiert wird. Beispiele: XS (Typ=S), XJ (Typ=J).

7. Kommandos

```
dd.mm.yy    hh:mm:ss    HOST: ........   TSN: ....             USER-ID: ........
COMMAND :
LASTP TYP FILENAME = $TSOS                                AGE PASS LACC  ACTION
   30 *I  xxxxxxxxxxxx  . . . . . . . . . . . . . . .  143 *      62 :
   77 *P  xxxxxxxxxxxx  . . . . . . . . . . . . . . .  143 *      62 :
    9 *P  xxxxxxxxxx  . . . . . . . . . . . . . . . .  143 *      62 :
   49 *P  xxx . . . . . . . . . . . . . . . . . . . .    4 . . .   2 :
   61 *I  xxxxxxxxxxxxxxxx  . . . . . . . . . . . . .    4 . . .   2 :
   10 *S  xxxxxxxxxxxxxxxx  . . . . . . . . . . . . .    4 . . .   4 :
   73  S  xxxxxxxxxxxxxxxxxxx . . . . . . . . . . . .    4 . . .   2 :
   29  S  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .    4 . . .   2 :
    0  N  xxxxxxxxxxxxxxxxxxxxx . . . . . . . . . . .      . . .   0 :
  245 *P  xxxxxxxxxxx . . . . . . . . . . . . . . . .    4 . . .   2 :
   49  P  xxxxxx  . . . . . . . . . . . . . . . . . .    2 . . .   0 :
  237 *P  xxxxxxxxxxxxx . . . . . . . . . . . . . . .    4 . . .   4 :
   12  S  xxxxxxxxxxxxxxx . . . . . . . . . . . . . .    8 . . .   7 :
    1  S  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .   45 . . .   4 :
    1  S  xxxxxxxxxxxxxxxxxxxx  . . .  . . . . . . . .  42 . . .   4 :
   14 *P  xxxxx . . . . . . . . . . . . . . . . . . .  290   RW   62 :
    9 *P  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .  148    W   53 :
    9  P  xxxxxxxxxxxxxxxxx . . . . . . . . . . . . .  148 *       1 :
*TAPE  S  xxxxxxxxxxx . . . . . . . . . . . . . . . .    4 . . .   4 :
   11 *P  xxxxxxxxxxxxxxxxxxxxxxx . . . . . . . . . .  148 *ERW   62 :
List continues   (P=1/T=187/H=0)          For Help:  ? / ?cmd / ?all / ??
```

Allgemeine Bemerkungen zu Kommandos

Über Kommandos - einzugeben im Feld COMMAND der Dateienliste - werden Funktionen von CFS angefordert, die nicht in unmittelbarem Zusammenhang mit einem bestimmten, in der Dateienliste aufgeführten Datenobjekt stehen, z.B. das Verschieben des Sichtfensters innerhalb der Dateienliste, Suchen eines Eintrags in der Dateienliste, usw.

Über Kommandos werden auch Funktionen von CFS aufgerufen, die nicht in Zusammenhang mit der angezeigten Dateienliste stehen, z.B. Aufrufen von Softwarekomponenten als Unterprogramm von CFS (EDT, EDOR, LMS, PERCON, ...), Eröffnen von Nebenprozessen (LOGON auf beliebige Benutzerkennungen, Anschluß an UTM- und DCAM-Anwendungen; ausführliche Beschreibung siehe Seite 337 und folgende "Connections").

Kommandos zum Anzeigen und Verändern von Datenobjekten sind ebenfalls im Feld COMMAND einzugeben. Nähere Einzelheiten siehe Seite 289 und folgende "CFS-Display/Editor".

Im Kommandofeld von CFS können schließlich auch allgemeine Programm-Modi gesetzt bzw. verändert werden. Siehe hierzu Seite 465 und folgende "Parameter ändern".

K-/F-/DUE-Tasten

Die K1-Taste bewirkt in CFS einen Rücksprung in die darüberliegende hierarchische Ebene

	LST		NP		END|*	
Display-Modus	--->	Dateienliste	--->	Selektionsmaske	--->	CFS-Ende
	K1		K1		K1	

Durch Betätigen der K1-Taste wird ferner die Anzeige der Help-Informationen, sowie die weitere Abarbeitung der I-/IM-Actions beendet.

Die F2-Taste bewirkt in der Dateienliste, daß alle angezeigten Dateinamen zum Zwecke des Umbenennens (Rename) überschreibbar gemacht werden.

Die F2-Taste im Display-Modus bewirkt, daß der Inhalt der angezeigten Datei überschreibbar gemacht wird (implizites Modify-Kommando).

Mit der F3-Taste wird der aktuelle Bildschirm in eine Hardcopy-Datei protokolliert.
Über ein File-Kommando mit LINK=HARDCOPY kann eine Hardcopy-Datei mit einem vorgegebenen Namen zugewiesen werden.

Mit der Shift ENTER-Taste (DUE2) wird die Anzeige des Kommandogedächtnisses angefordert. Weitere Informationen hierzu finden Sie im Abschnitt "Gedächtnis der eingegebenen Kommandos" auf Seite **211**.

Blanks in Kommandos

Kommandos können formatfrei eingegeben werden. Ein Kommandoscanner sorgt dafür, daß ein bis drei aufeinanderfolgende Blanks aus Kommandos entfernt werden. Intern arbeitet CFS mit den komprimierten Kommandos. Bei BS2000-Kommandos findet keine Komprimierung statt.

Verkettung mehrerer Kommandos

Im Kommandofeld von CFS können mehrere Kommandos nacheinander zur Ausführung gebracht werden. Die einzelnen Kommandos werden durch das Separatorzeichen Semikolon ";" getrennt. Zur besonderen Bedeutung des Separatorzeichens bei der Dateiselektion mit den Kommandos AL/NP, siehe Seite **256**.

Beispiele:

```
SPL;WAIT;LOCK
ADD FS.DO.PRINT,CFS.LIB.PLAM,J/DRUCK;/ER FS.DO.DRUCK
```

Das Separatorzeichen ";" kann für besondere Zwecke auch in ein anderes Zeichen umdefiniert werden. Das zugehörige Kommando lautet: SEP *x* (*x*: neues Separatorzeichen).

Eine ausführlichere Beschreibung des Kommandos SEP finden Sie auf Seite **481** (Parameter ändern).

* als Platzhalter für Dateinamen in Kommandos

* kann in Kommandos als Platzhalter für den Namen eines in der Dateienliste aufgeführten Datenobjekts verwendet werden. Der Name der gewünschten Datei, des Bibliothekselements, der Jobvariablen wird durch Eingabe von * im Action-Feld festgelegt. Noch im gleichen Dialogschritt kann das Symbol * in Kommandos als Platzhalter für den damit verbundenen Namen verwendet werden. Die so definierte Zuordnung Name <--> * bleibt bestehen, bis sie durch eine neue Zuordnung ersetzt wird.

Neben * können die Action-Codes *1, ..., *9 zum Merken weiterer Namen verwendet werden. Im Kommandofeld von CFS können damit die Namen von bis zu 10 verschiedenen Datenobjekten über entsprechende Platzhalter angesprochen werden.

Die Zuordnungen Name <--> *1, ..., *9 bleiben während eines CFS-Laufs bestehen (auch bei Selektion einer neuen Dateienliste). Für die Zuordnung * gilt standardmäßig das gleiche.
Durch eine Modifikation im CFS-Vorschaltprogramm CFSMAIN kann jedoch der folgende alternative Modus für * aktiviert werden: Die Namenszuordnung zu * wird gelöscht, nachdem sie im Kommandofeld zur Namenssubstitution verwendet wurde. Damit werden die Fälle von möglichen Fehlsubstitutionen von Zeichenfolgen wie *any, *priv, *std, *lib usw. im Kommandofeld verhindert.

Bei Elementen einer Bibliothek wird durch Angabe von * im Kommandofeld von CFS auf die vollständige Bezeichnung eines Bibliothekselements Bezug genommen. Je nach Kontext gehört dazu der Name der Bibliothek, der Name des Elements, der Elementtyp (PLAM) sowie die Versionsbezeichnung. In bestimmten Fällen ist es jedoch wünschenswert, nur den Elementnamen zu berücksichtigen und den Namen der Bibliothek, die Versionsnummer und den Elementtyp zu ignorieren. In diesen Fällen ist in den Kommandos *E anstelle von * als Platzhalter zu verwenden.

Im Zusammenhang mit manchen BS2000-Kommandos wird das Zeichen * im Kommandofeld nicht als Kürzel für einen Namen verwendet, z.B. /DO CFS.DO.TRANSFILE,(*CFS*). In diesem Prozeduraufruf ist '*CFS*' ein Parameter für die DO-Prozedur. Eine Ersetzung durch einen mit * verknüpften Namen ist hier nicht gewünscht. Um die Ersetzung von * durch einen Namen aus der Dateienliste zu verhindern, kann das Kommando CL* (Clear *-Action-Code) verwendet werden. CL* bewirkt, daß die Zuordnung * <--> Name aufgelöst wird. * kann danach als normales Zeichen eingegeben werden, ohne daß eine Ersetzung durch einen Namen erfolgen würde.

Bei den Kommandos NP* und AL* wird in keinem Fall die Ersetzung des Zeichens * durch einen entsprechend markierten Namen ausgeführt. Hier hat * eine besondere Bedeutung als Parameter für das entsprechende Kommando:

NP* Neuselektion der Dateienliste unter Zuhilfenahme der zuletzt angegebenen Selektionskriterien.

AL* Verlängern der aktuellen Dateienliste, wobei in die Selektionsmaske verzweigt wird und die zuletzt eingetragenen Auswahlbedingungen wieder angezeigt werden.

Die Eingabe von * in Spalte 1 des Kommandofeldes beendet CFS.

Jobvariablenersetzung in Kommandos

Wird in einer Kommandoeingabe das Konstrukt &(*jv-name*) angegeben, so wird dieses von CFS durch den momentanen Inhalt der Jobvariablen ersetzt.

Beispiele:

```
/CAT *,RDPASS=&(CFS.PASSW),STATE=U
OC1/LOGON TSOS,ADMINSTR,&(PSW)
```

Gedächtnis der eingegebenen Kommandos

CFS führt intern eine Tabelle (8192 Byte), in der alle eingegebenen Kommandos eingetragen werden. Auf dieses "Kommando-Gedächtnis" kann der Benutzer auf zwei verschiedene Arten zugreifen:

a) sequentiell:

Durch Betätigen der Taste Shift ENTER (DUE2). Falls das Kommandofeld leer gelassen wurde, wird das letzte, vorletzte, usw. Kommando angezeigt.

b) assoziativ:

Durch Eingabe eines Strings im Kommandofeld und Betätigen der Taste Shift ENTER (DUE2) wird das letzte, vorletzte, usw. Kommando angezeigt, das mit der Zeichenfolge '*string*' beginnt.

Bei Angabe eines Strings unter Voranstellung des Wildcardzeichens * (**string*) wird das letzte, vorletzte, usw. Kommando angezeigt, das die Zeichenfolge '*string*' an beliebiger Stelle enthält.

Durch wiederholtes Betätigen der Taste Shift ENTER kann der Benutzer beliebig weit in der Kommandohistorie zurückgehen.

Beispiele:

`COMMAND :` OC Shift ENTER (DUE2)
zeigt das zuletzt eingegebene Open Connection-Kommando an.

`COMMAND :` *FILE Shift ENTER (DUE2)
zeigt das letzte Kommando, das die Zeichenfolge 'FILE' enthält.
z.B. /FILE ... oder ONX/FILE ...

Inhalte aus Kommandogedächtnis in P-Tasten laden

Ein am Bildschirm angezeigter Eintrag aus dem Kommandogedächtnis kann in eine frei bestimmbare P-Taste geladen werden. Hierzu ist die Bezeichnung der P-Taste der Eingabe voranzustellen: 'P*x*:' bzw. 'P*x*#'. *x* ist hierbei die ein- bis zweistellige Nummer der gewünschten P-Taste. Der am Bildschirm angezeigte Eintrag aus dem Kommandogedächtnis ist mit ENTER zu senden. Die Eingabe wird damit in die angegebene P-Taste geladen. Die Ausführung der Eingabe als Kommando wird in diesem Fall unterdrückt.

Wird der Eingabe aus dem Kommandogedächtnis die Zeichenfolge *EM bzw. *EM*DUE angefügt, so wird der String in der P-Taste mit ◄ bzw. ◄ ENTER abgeschlossen.
Der String in der P-Taste wird automatisch mit ◄ ENTER abgeschlossen, falls das Prefix 'P*x*#' anstelle von 'P*x*:' angegeben wird.

Beispiel:

Aus dem Kommandogedächtnis wird die Eingabe `/STA 7813` am Bildschirm angezeigt. Durch Modifizieren der Eingabe nach `P12:/STA 7813*DUE` und Absenden mit ENTER wird /STA 7813 in die Taste P12 übertragen. Aufgrund von *DUE wird bei Betätigung von P12 automatisch eine Datenübertragung ausgelöst.

Hinweis:

Aus Datenschutzgründen ist ein nachträgliches Entfernen von Eingaben aus dem Kommando-gedächtnis möglich: Mit der Taste Shift ENTER (DUE2) wird die zu entfernende Eingabe im Kommando-Gedächtnis aufgesucht. Durch Betätigen der K3-Taste wird diese Eingabe dann aus dem Kommando-Gedächtnis gelöscht.

24-zeilige Ausgabe des Kommandogedächtnisses

Durch Voranstellen von '--' vor den zu suchenden Begriff wird eine ganze Bildschirmseite (max. 24 Zeilen) mit Eingaben angezeigt, die den gesuchten Begriff enthalten.

Es bestehen folgende Möglichkeiten der Eingabe:

`COMMAND :` -- Shift ENTER (DUE2)
zeigt die 24 letzten eingegebenen Kommandos.

`COMMAND :` --*string* Shift ENTER (DUE2)
zeigt die 24 letzten Kommandos, die mit '*string*' beginnen.

`COMMAND :` --**string* Shift ENTER (DUE2)
zeigt die 24 letzten Kommandos, die an beliebiger Stelle die Zeichenfolge '*string*' enthalten.

Durch weiteres Betätigen der Taste Shift ENTER (DUE2) wird die nächste Bildschirmseite mit Eingaben des gesuchten Typs ausgegeben usw.

Hinweise:

Auch bei der seitenweisen Anzeige des Kommandogedächtnisses können ein oder mehrere Zeilen aus dem Gedächtnis in verschiedene, frei wählbare P-Tasten übertragen werden. Die Vorgehensweise ist dabei die gleiche wie oben beschrieben ("Inhalte aus Kommandogedächtnis in P-Tasten laden").

Für weitere Informationen zum Thema Kommandogedächtnis siehe Seite **252**/**281**: Load Memory/Save Memory), Seite **478**: Log Input in Memory und Seite **469**: Automatic Memory Compression.

In Zusammenhang mit dem Kommandogedächtnis ist auch das Kommando KC (Keep Command) von Bedeutung. Mit diesem Kommando wird CFS in einen Modus versetzt, in dem das Kommandofeld nach der Eingabe nicht gelöscht wird. Das zuletzt eingegebene Kommando bleibt damit solange im Kommandofeld stehen, bis es durch Blanks oder durch ein anderes Kommando überschrieben wird.

Rechnen und Konvertieren

CFS-Kommandos, die mit dem Zeichen '=' beginnen, werden von einem internen Rechen- und Konvertierungsmodul ausgewertet. Das Ergebnis wird in der Systemzeile (Zeile 25) ausgegeben. Es stehen folgende Funktionen zur Verfügung:

Arithmetische und logische Operationen:

Es werden die vier Grundrechenarten + - * /, sowie %-Rechnung und die logischen Operationen .AND., .OR., .XOR. angeboten. Als Operanden können hexadezimale oder dezimale Zahlen angegeben werden. Die Rechenausdrücke sind in folgendem Format anzugeben:

=*zahl* [*op zahl*] ...

zahl Dezimal oder Hexadezimalzahl.
Zahlen, die nur Ziffern 0 bis 9 enthalten, werden als Dezimalzahlen interpretiert. Enthält eine Zahl die Zeichen A bis F, so wird sie als Hexadezimalzahl interpretiert. Das Anhängen des Zeichens **H** oder **X** an eine Zahl in dezimaler Schreibweise (z.B. 10H) bewirkt, daß diese Zahl als Hexadezimalzahl interpretiert wird.

op Arithmetischer oder logischer Operator. Es stehen die folgenden Operatoren zur Verfügung:

+ Addition
- Subtraktion
* Multiplikation
/ Division
% %-Rechnung
.+. Addition ohne Berücksichtigung des Vorzeichens (Adressrechnung)
.-. Subtraktion ohne Berücksichtigung des Vorzeichens (Adressrechnung)
.AND. logische UND-Verknüpfung
.OR. logische ODER-Verknüpfung
.XOR. logisches Exklusiv-ODER

Die Abarbeitung der einzelnen Operationen erfolgt rein linear. Beispiel: =10+10*2 ergibt als Ergebnis 40 und nicht 30, wie dies nach der Regel "Punkt vor Strich" zu erwarten wäre. Gleiches gilt für alle logischen Operationen.

Beispiele:

```
=16+80H
```
In der Systemzeile wird als Ergebnis der Addition der Wert 144 in dezimaler und hexadezimaler Darstellung (90H) ausgegeben.

```
=7FFFFD45.+.FFF
```
Als Ergebnis der logischen Addition wird der Wert 80000D44 ausgegeben. Die arithmetische Addition =7FFFFD45+FFF würde zu der Fehlermeldung 'result too big ...' führen, da die größtmögliche positive Hexadezimalzahl 7FFFFFFF überschritten wird (Festpunktüberlauf).

```
=1200%14+1200%85
```
Der angegebene Ausdruck wird in folgender Reihenfolge berechnet:
14% von 1200=168, 168+1200=1368, 85% von 1368=1162.

```
=7C
```
Für die angegebene Zahl wird die dezimale und hexadezimale Darstellung ausgegeben: 7C= 124 (7C)

```
=1.OR.2.OR.4.OR.8
```
Die vier Hexadezimaloperanden werden mit einem logischen Oder verknüpft. Als Ergebnis wird der Hexadezimalwert F ausgegeben.

```
=AFFE.AND.5001H
```
Die zwei Hexadezimaloperanden werden mit einem logischen Und verknüpft. Als Ergebnis wird der Hexadezimalwert 0H ausgegeben.

```
=7FFE.XOR.FF0
```
Die zwei Hexadezimaloperanden werden mit einem logischen Exklusiv-Oder verknüpft. Als Ergebnis wird der Hexadezimalwert 700EH ausgegeben.

Character- und Hexadezimalkonvertierung:

In hexadezimaler Schreibweise angegebene Strings werden in Character Darstellung konvertiert. In Character Darstellung angegebene Strings werden in hexadezimale Form konvertiert. Die zu konvertierenden Strings sind in folgendem Format anzugeben:

=X'*string*' | =[C]'*string*'

Bei der Konvertierung in Character-Format werden nicht darstellbare Zeichen als '?' angezeigt.

Beispiele:

`=X'C1C442'`
In der Systemzeile wird für X'C1C442' die Character-Darstellung C'AD?' angezeigt. Im Characterformat nicht darstellbare Zeichen werden durch '?' repräsentiert.

`='AFFE'`
In der Systemzeile wird für C'AFFE' die hexadezimale Darstellung X'C1C6C6C5' angezeigt.

Hilfe anfordern

? | HELP Menue aller Kommandos / globales Help-Menü anzeigen.

?*cmd* Es wird die Beschreibung zu dem gewünschten Kommando angezeigt.

?NEWS Änderungsprotokoll anzeigen.

?TAB Tabellenwerk anzeigen. In einem Menue wird eine Auswahl verschiedener Tabellen angeboten. Z.B. EBCDIC-/ASCII-Zeichen, Umsetzung, 9750-Gerätesteuerzeichen, DCAM-Returncodes, Program-Interrupt Codes, SVC-Nummern mit den zugehörigen Makros.

?USER Benutzereigene Helpdatei anzeigen. Für ausführliche Informationen zu benutzereigenen Helpdateien siehe Seite 553.

CFS-Programmbeendigung

***** Beendigung von CFS. Anstelle von * kann auch das Kommando END eingegeben werden. * bzw. END können auch in der Selektionsmaske eingegeben werden, um das Programm zu beenden.
* kann im Kommandofeld von CFS auch als Platzhalter für den Namen eines Datenobjekts verwendet werden (siehe Seite 209).
In CFS-Prozeduren (siehe Seite 496) kennzeichnet * eine Kommentarzeile und kann somit nicht zur Beendigung von CFS verwendet werden. In diesem Fall ist CFS durch das Kommando END zu beenden.

Sichtfenster in Dateienliste verschieben

+ | - Ausschnitt der Dateienliste um einen Bildschirm weiter zum Ende/Anfang verschieben. Das leere Kommandofeld - abgesendet mit ENTER - hat die gleiche Wirkung wie das Kommando '+': Es wird um einen Bildschirm weitergeblättert.

++ | -- Ausschnitt der Dateienliste zum Ende/Anfang verschieben.

+*n* | -*n* Ausschnitt der Dateienliste um *n* Zeilen weiter zum Ende/Anfang verschieben.

P*n* Ausschnitt der Dateienliste auf den *n*-ten Eintrag positionieren. *n* entspricht hierbei der Positionsangabe in der letzten Bildschirmzeile (P=*n*/T=*m*/H=*p*).

Breakpoint / BS2000-Kommando ausführen

/ Break: In BS2000-Kommandomodus verzweigen.
Rückkehr in das Programm CFS durch das BS2000-Kommando **/R** [esume].

Eine evtl. angezeigte Display-Datei wird von CFS geschlossen, bevor der Wechsel in den BS2000-Kommandomodus ausgeführt wird. Durch Anwendung der K2-Taste gelangt der Benutzer aus dem Programm ebenfalls in den Kommandomodus. Die Display-Datei wird dabei jedoch nicht geschlossen.

/cmd BS2000-Kommando *cmd* ausführen.

% *cmd* AID-Kommando ausführen.

Hinweise:

Bei /CAT ... wird der Parameter ',STATE=U' ergänzt, falls im Kommando nichts weiter angegeben wurde.

Beim BS2000-Kommando /DO ... werden die Parameter für die DO-Prozedur auf Vollständigkeit geprüft. Falls einem in der PROC-Anweisung vorkommenden Parameter noch kein Wert zugewiesen ist, wird von CFS eine Parameter-Maske ausgegeben. In ihr erscheinen alle in der PROC-Anweisung definierten Parameter zusammen mit den zugehörigen Werten.

Die Parameter-Maske kann auch unvollständig ausgefüllt abgesendet werden. In diesem Fall erhalten die leeren Parameter keinen Wert und werden bei Bedarf vom BS2000 im Dialog angefordert (Prompting). In der Parameter-Maske ausgefüllte Parameter werden nicht mehr gepromptet. Die Prüfung der DO-Parameter und die Ausgabe der Parameter-Maske kann unterdrückt werden durch den Verarbeitungsmodus NDP (No Do Parameters).
Die Ausführung eines /DO-Kommandos führt zur Beendigung von CFS.

Ausführen der Actions

A [*opt*] CFS beginnt mit der Ausführung der gesammelten und nicht sofort ausgeführten Actions wie P (Print), E (Erase) und X (Variable Action ONX...).
Zusammen mit dem Kommando A kann als wahlweiser Zusatz eine Eingabe für die Terminierungsabfrage mitgegeben werden. Der Zusatz *opt* bewirkt, daß die Terminierungsabfrage unterdrückt und die entsprechende Programmverzweigung ausgeführt wird. Folgende Varianten sind möglich:

opt * | N | NP *param* | U | UK | T

* Actions ausführen und danach die Selektionsmaske mit dem zuletzt eingetragenen Inhalt anzeigen.

N Actions ausführen und danach in die leere Selektionsmaske verzweigen.

NP *param* Actions und danach das angegebene NP-Kommando (Dateiselektion) ausführen. Die Eingabemöglichkeiten für *param* sind beim Kommando NP auf Seite 256 beschrieben.

U Actions ausführen und danach die bisherige Dateienliste in der aktualisierten (upgedateten) Form anzeigen.
Bei Angabe von AU wird nicht auf den Anfang der Dateienliste, sondern auf den zuletzt angezeigten Ausschnitt positioniert. In der Dateienliste eingetragene Action-Codes X für Variable Actions werden nicht gelöscht.

UK wie U, jedoch wird eine zuvor definierte Variable Action für die weitere Verwendung beibehalten.

UE Es werden nur die Einträge der letzten Dateienliste angezeigt, für die die Variable Action nicht fehlerfrei ausgeführt werden konnte. Die Einträge der Dateienliste, für die keine Variable Action oder die Variable Action ohne Fehler ausgeführt wurde, werden intern mit dem Action-Code '-' unsichtbar gemacht. Die vollständige Dateienliste kann mit dem Kommando YANK wieder angezeigt werden. Ohne Eingabe von UE werden fehlerhaft ausgeführte Variable Actions in der rechten Fehlerspalte der Dateienliste mit 'ERROR' gekennzeichnet.

T Actions ausführen und danach CFS beenden.

Für eine ausführliche Beschreibung der möglichen Antworten auf die Terminierungsabfrage siehe Seite **545** "Terminierungsabfrage".

Datei in Bibliothek aufnehmen

ADD *datei* [, *bibliothek*] [, *element*]

Variante 1: **kein Elementtyp angegeben**

Datei als Element in eine Bibliothek aufnehmen.
Die Angabe des Bibliotheksnamens kann entfallen, sofern eine der folgenden Bedingungen erfüllt ist:

a) Die Dateienliste besteht aus Elementen einer Bibliothek und die angegebene Datei soll in die gleiche Bibliothek aufgenommen werden.

b) in einem vorhergehenden ADD-Kommando wurde ein Bibliotheksname angegeben und der aktuelle ADD geht in die gleiche Bibliothek.

Die Angabe des Elementnamens ist optional. Falls kein Elementname angegeben wurde, wird dieser nach folgender Regel aus dem Dateinamen gebildet (abhängig von der Art der Bibliothek):

FMS	Elementname = Dateiname
LMS (ISAM)	Elementname = erste 8 Zeichen der letzten Teilqualifizierung
PLAM	Elementname = Dateiname. Der Elementtyp für PLAM wird in diesem Fall nachgefordert.

ADD *t*/*datei* [, *bibliothek*] [, *element*] | **ADD** *datei* [, *bibliothek*] [, *t*/*element*]

Variante 2: **Angabe eines Elementtyps**

Datei als Element in eine Bibliothek aufnehmen.
Diese Variante des ADD-Kommandos ist nur zulässig bei PLAM-Bibliotheken.
Die zwei oben dargestellten Ausprägungen der Variante 2 (Angabe des Elementtyps vor dem Datei- bzw. vor dem Elementnamen) sind gleichwertig.

t/ Elementtyp, unter dem die aufzunehmende Datei als Element in der PLAM-Bibliothek abgelegt werden soll.

Hinweise:

Falls eine Bibliothek des im ADD-Kommando angegebenen Namens noch nicht existiert, so wird der Benutzer gefragt, von welcher Art die von CFS neu einzurichtende Bibliothek sein soll:

"Create FMS- / LMS- (ISAM) / PLAM-Library / ZIP / None ? (F/M/L/P<2>/Z/N)"

Die Eingabe N sollte verwendet werden, wenn sich der Benutzer im Bibliotheksnamen vertippt hat und CFS nun eine Bibliothek mit diesem Namen einzurichten versucht.

Die Eingabe P2 bewirkt, daß eine PLAM-Bibliothek mit BLKSIZE=(STD,2) eingerichtet wird. Doppelt geblockte Bibliotheken bieten bei Einsatz von PLAM ab Version 2.0 deutliche Performancevorteile gegenüber Bibliotheken, die mit (STD,1) geblockt sind. Es ist jedoch zu beachten, daß dieses Bibliotheksformat nicht abwärtskompatibel ist und mit LMS/PLAM in einer Version kleiner als 2.0 nicht verarbeitet werden kann.

Mit der Eingabe Z wird ein neues ZIP-Archiv erzeigt.

Besonderheiten bei Delta-Speicherung

Delta-Speicherung bedeutet, daß verschiedene Versionen des gleichen Elements in der Form abgelegt sind, daß jeweils nur die geänderten Sätze bezüglich der Vorgängerversion gespeichert werden. Durch die Delta-Technik kann im Vergleich zur Vollspeicherung der Elemente eine bedeutende Reduzierung des Speicherplatzes erreicht werden.

Bei der Aufnahme einer Datei in eine PLAM-Bibliothek ist bezüglich Delta-Speicherung folgendes zu beachten:

Existiert noch kein Element des im ADD-Kommando angegebenen Namens und Typs in der Bibliothek, so wird das neue Element in jedem Fall als Voll-Element, d.h. in nicht delta-gespeicherter Form angelegt.

Existiert bereits ein gleichnamiges Element in der PLAM-Bibliothek, so wird die bereits bestehende Speicherungsform beibehalten. Dies bedeutet:

a) Liegt ein Element gleichen Namens und Typs in der Zielbibliothek als Voll-Element (d.h. in nicht delta-gespeicherter Form) vor, so wird das bisherige Element durch die neu aufzunehmende Datei überschrieben, sofern nicht verschiedene Versionen vorliegen.

b) Existiert bereits ein deltagespeichertes Element gleichen Namens in der Zielbibliothek, so wird die aufzunehmende Datei als nächsthöhere Version an den bestehenden Delta-Baum angehängt. Das alte Element wird dadurch nicht überschrieben und steht weiterhin zur Verfügung.

Zum Thema Delta-Speicherung in PLAM-Bibliotheken siehe auch Seite **44**, **82**, **119** und **125**.

Es können Dateien beliebigen Formats in FMS-Bibliotheken aufgenommen werden. Insbesondere können auch PAM-Dateien (gebundene Programmphasen) und Dateien mit RECFORM=F,BLKSIZE=(STD,*n*) oder ISAM-Dateien. Die so aufgenommenen Elemente können mit dem Programm FMS jedoch nicht bearbeitet werden. Eine Selektion derartiger Elemente ist nur über CFS möglich.

Bei Aufnahme von Elementen in eine PLAM-Bibliothek prüft CFS den Elementtyp auf LMS-Kompatibilität. Es sind zwei Varianten möglich:

a) nicht LMS-kompatible Typbezeichnungen werden von CFS abgewiesen.

b) bei nicht LMS-kompatiblen Elementtypen wird der Benutzer gefragt, ob er ein Element unter dem angegebenen Typ in die Bibliothek aufnehmen möchte.

Die Kommandos zur Steuerung der Typ-Überprüfung lauten:

CT Check Types: Nur LMS-kompatible Elementtypen zulassen.

CTQ Check Types Query: Bei nicht LMS-kompatiblen Elementtypen Benutzer fragen. (Standard).

NCT No Check Types: Keine Typ-Überprüfung.

Mit LMS verträgliche Elementtypen für PLAM-Bibliotheken sind:

C Coreimage (mit TSOSLNK gebundene Programme)
D Document/Data
H von H-Assembler erzeugte Daten
J JCL (DO-/Enter-Prozeduren)
L vom Binder des BS2000 erzeugte Daten
M Macro
P Print (mit SPACE=E ausdruckbare Listdateien)
R Relocatable (Objektmodule)
S Source
X Extra

Für weitere Informationen zum Thema Aufnahme von Dateien in Bibliotheken siehe auch Seite **114**: Variable Action ONXADD.

Automatic Documentation

ADOC | NADOC

Automatic Documentation File / No Automatic Documentation File

ADOC

Bei jeder neuen Zusammenstellung einer Dateienliste (Selektionsmaske/NP-Kommando) wird die alte Dateienliste in eine temporäre Datei gesichert.
Standard: ADOC.

Hinweise:

Der Name der Zwischendatei lautet #CFS.SAVELIST.*jobname* bzw. #CFS.SAVE LIST.*acct-nr* (*acct-nr* nur, falls im LOGON-Kommando kein Jobname angegeben wurde). Das standardmäßige Prefix #CFS.SAVELIST. kann vom CFS-Administrator auch modifiziert werden.

Durch Eingabe von RL (Restore List) im Feld FILENAME-SELECT wird die letzte Dateienliste wieder angezeigt.

Mit dem Kommando RL kann die Anzeige zwischen der letzten und der aktuellen Dateien-/Bibliotheksliste alterniert werden. Das Kommando RL erlaubt somit einen raschen Wechsel zwischen zwei verschiedenen Dateien-/Bibliothekslisten.

Dateienliste: Anzeige der letzten Veränderung als Anzahl von Tagen

AGE

Altersangaben, insbesondere die Angaben in der AGE-Spalte der Dateienliste, werden als Anzahl von Tagen dargestellt. Durch das Kommando DATE oder NAGE kann das Alter auch in Form einer Datumsangabe angezeigt werden. Anstelle von AGE kann auch das Kommando NDATE angegeben werden.
Standard: AGE.

Versionshochzählung bei ADD auf bestehende Elemente in LMS-/ PLAM-Bibliotheken

AI | NAI

Add with Increment of version/No Add with Increment of version.

AI

Automatische Versionserhöhung bei ADD in LMS-Bibliotheken (LMS-ISAM/ PLAM): Falls ein Element gleichen Namens in der Bibliothek bereits existiert, wird die Versionsbezeichnung des neu aufzunehmenden Elements um 1 erhöht.

NAI

Beim ADD in LMS-Bibliotheken wird, falls bereits ein Element gleichen Namens in der Bibliothek existiert, dasjenige mit der höchsten Versionsnummer überschrieben. Dies gilt jedoch nicht, falls es sich um ein in Delta-Form gespeichertes Element in einer PLAM-Bibliothek handelt. In diesem Fall wird auch im NAI-Modus eine automatische Erhöhung der Versionsnummer um 1 vorgenommen.
Standard: NAI.

Dateienliste durch neue Selektion ergänzen

AL

Append List. CFS verzweigt in die Selektionsmaske. Die aktuelle Dateienliste kann durch weitere Datenobjekte des gleichen Typs - Dateien der gleichen oder einer anderen Benutzerkennung, Elemente aus der gleichen Bibliothek - ergänzt werden. Es ist kein Mix zwischen Datenobjekten verschiedenen Typs oder zwischen Elementen aus verschiedenen Bibliotheken möglich.

AL *param*

Das Feld FILENAME-SELECT und andere Felder der Selektionsmaske werden mit dem durch *param* angegebenen Inhalt gefüllt und die Dateienliste wird um die neu selektierten Dateien ergänzt. Das manuelle Eintragen der gewünschten Auswahlbedingungen in der Selektionsmaske entfällt bei dieser Variante des AL-Kommandos.

param

Für *param* können die gleichen Angaben gemacht werden wie im Kommando NP (NP *param*) Seite **256**.

Beispiel: `AL XYZ;AG=0`
Die Dateienliste wird um alle Elemente aus der bereits selektierten Bibliothek erweitert, die die Zeichenfolge 'XYZ' in ihrem Namen enthalten und die ein Alter von 0 Tagen besitzen.

AL %*name* | %? [,*jrsave*] [,?] [;*param*]

Das Feld FILENAME-SELECT und andere Felder der Selektionsmaske werden mit dem in dem Parameterset gespeicherten Inhalt gefüllt und die Dateienliste wird mit den gefundenen Dateien ergänzt.

%*name* Name des Parametersets mit der Selektionsbedingung.
%? Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

jrsave Name der JRSAVE-Datei, in der der angegebene Parameterset enthalten ist. Standard: CFS.JRSAVE.*xxxx*. Näheres siehe Seite 565.

? Die Selektionsmaske wird mit dem Inhalt des Parametersets gefüllt und dem Benutzer angezeigt. Standardmäßig wird die Selektion sofort ausgeführt.

param Mit Ausnahme von *filename-sel* können für *param* die gleichen Angaben gemacht werden wie bei AL *param* (siehe oben). Die in *param* angegebenen Werte überschreiben die entsprechenden Werte des Parametersets.

AL* Diese Variante des AL-Kommandos bewirkt, daß in die Selektionsmaske verzweigt wird, wobei diese bereits mit den zuletzt eingegebenen Auswahlbedingungen vorbesetzt ist (Unterschied zu NP*).

Hinweise:

Das Abspeichern einer Selektionsbedingung in einem Paramset erfolgt mit dem Kommando SP %*name*. Näheres hierzu auf Seite 283.

Falls die aktuelle Dateienliste aus Elementen einer Bibliothek besteht und mit dem AL-Kommando um weitere Elemente aus der gleichen Bibliothek verlängert werden soll, so ist die folgende Abkürzung möglich: Der zur Auswahl der Bibliothekselemente notwendige Vorspann LMS *bibl* / FMS *bibl* / LMR *bibl* kann weggelassen werden, da er von CFS ergänzt wird.

Zum Thema "Dateien der Dateienliste hinzufügen" wird auch auf das Kommando I (Include) hingewiesen. Mit dem Kommando NP (New Parameters) wird eine neue Dateienliste aufgebaut; die alte wird vergessen.

Der Arbeitsschritt des Erweiterns der Dateienliste kann in vielen Fällen eingespart werden durch geeignete Mehrfach-Auswahlbedingungen bezüglich des Dateinamens (Und-, Oder-Bedingungen, Wildcard-Syntax). Ausführliche Informationen hierzu Seite 51.

Allocated Space als Merkmal für Dateigröße

ALLOC In der ersten Spalte der Dateienliste wird als Maß für die Größe einer Datei der Wert für Allocated Space (zugewiesene PAM-Seiten) angezeigt.

Standardmäßig wird für die Dateigröße die Anzahl der beschriebenen PAM-Seiten (LASTP) angezeigt.

Kommandogedächtnis automatisch komprimieren

AMC | NAMC Automatic Memory Compress/No Automatic Memory Compress.

AMC Jedes der CFS-internen Gedächtnisse (FILENAME-SELECT, COMMAND, Connection 0, ..., 9) besteht aus einer Speichertabelle von 4096 Bytes, in die die Tastatureingaben der Reihe nach eingetragen werden. Bei Erreichen des Endes einer dieser Tabellen werden die Tabelleneinträge am Anfang zyklisch überschrieben. Um dieses Überschreiben der im Kommandogedächtnis festgehaltenen früheren Eingaben zu verzögern und noch mehr Freiplatz zu gewinnen, wird intern ein Memory Compress durchgeführt. Der Memory-Compress bewirkt, daß gleiche Eingaben aus dem Gedächtnis eliminiert werden und dadurch mehr freier Platz in der Tabelle zur Verfügung steht. Nach der Komprimierung entspricht die Reihenfolge der Eingaben im Kommandogedächtnis nicht mehr der tatsächlichen Reihenfolge der getätigten Eingaben. Standard: AMC.

Zum Thema Kommandogedächtnis siehe auch CM (Compress Memory, Seite 223), SM (Save Memory, Seite 281) und LM (Load Memory, Seite 252).

Adressierungsmodus abfragen / ändern

AMODE Es wird der Adressierungsmodus ausgegeben, in dem CFS gegenwärtig läuft.

AMODE 31|24 Der Adressierungsmodus (Run-Mode) von CFS wird auf XS-31 oder NXS (24-bit Modus) gesetzt.
Standard beim Start von CFS: AMODE24. Um den Standard von AMODE24 auf AMODE31 zu setzen, ist im Initialisierungsmodul CFSMAIN die Anweisung AMODE ANY in AMODE 31 zu ändern.

Hinweise:

Im AMODE31 wird bei der Ausführung verschiedener CFS-Kommandos bzw. Action-Codes (z.B. EDT, COMP) der benötigte virtuelle Speicher im oberen Adressraum angefordert. Dies bietet den Vorteil, daß die 16MB-Grenze nicht existiert und somit nahezu beliebig viel Speicher angefordert werden kann.

Beim Umschalten von AMODE31 auf AMODE24 is zu berücksichtigen, daß es im nachfolgenden Programmlauf zu Fehlersituationen kommen kann, die dadurch bedingt sind, daß im XS-Bereich angeforderter Speicher nicht mehr adressiert werden kann.

Drucker für Action-Code PDxx zuweisen

APD *device* Vordefinieren bzw. Ändern eines Druckerdevice-Namens, der in der PDFILE mit einem Mnemo-Code und dem virtuellen Device-Namen '????????' definiert ist. Falls einem virtuellen Device-Namen '????????' mit APD noch kein realer Drucker zugewiesen wurde, so wird der Druckername beim ersten Aufruf des entsprechenden Mnemo-Codes angefordert. Für weitere Informationen siehe CFS-Handbuch für Systemverwalter, Abschnitt "PDFILE".

BCMAP-Savedatei in Prozedurformat übersetzen

BCMAP *file1*,*file2* [, R|P]

Mit dem Systemverwalter bzw. Konsol-Kommando BCMAP FUNCT=SAVE,FILE=*file1* können die aktuellen BCMAP-Zuordnungen in einer Datei gesichert werden. Die Zuordnungen werden in der Save-Datei in einem internen Format abgelegt, welches für den Benutzer nicht interpretierbar ist. Das CFS-Kommando BCMAP erzeugt aus einer BCMAP-Savedatei eine ausführbare Run-Datei oder eine DO-Prozedur. Die BCMAP-Anweisungen werden somit wieder in Klartext umgewandelt. Die von CFS erzeugte Ausgabedatei kann vom Benutzer geändert und z.B. unter einer anderen Umgebung zur Ausführung gebracht werden.

file1 Name der mit BCMAP FUNCT=SAVE,FILE=*file1* erzeugten Sicherungsdatei.

file2 Name der von CFS zu schreibenden Ausgabedatei.

R|P Die Ausgabedatei wird als Run-Datei (Standard) bzw. als BS2000 DO-Prozedur erzeugt.
Eine Run-Datei enthält Anweisungen ohne Fortsetzungszeilen in einer Länge bis zu 200 Bytes. In einer BS2000 Do-Prozedur sind die Anweisungen u.U. über mehrere Zeilen verteilt mit Fortsetzungszeichen '-' in Spalte 72.

Hinweise:

Das BCMAP-Kommando von CFS kann auch als "Stand-Alone" Funktion aufgerufen werden, ohne daß CFS geladen ist. Der Programmaufruf is in folgender Form anzugeben: /EXEC (BCMAP,$CFS.CFSLIB) .
Die Namen der Ein- und Ausgabedateien werden in diesem Fall von der Eingabequelle SYSDTA angefordert.

Monatskalender anzeigen

CAL [*mm.* [*yyyy*]]

Calendar. Es wird am Bildschirm der Kalender des angegebenen Monats angezeigt. Sind für bestimmte Tage des angezeigten Monats Termine vorhanden, so wird bei dem entsprechenden Tag ein "x" eingetragen (ein Termin), bzw. "xx" (mehrere Termine). Bei der Anzeige des gegenwärtigen Monats (CAL) wird der aktuelle Tag blinkend dargestellt.

Eine ausführliche Beschreibung des Kommandos finden Sie auf Seite **427** und folgende (Terminverwaltungssystem).

Dateinamen in voller Länge anzeigen

CFN | NCFN Complete Filename. Die am Bildschirm angezeigte Dateienliste enthält die Datei- und Jobvariablennamen in der vollständigen (FSTAT-kompatiblen) Form, d.h. es werden stets die Cat-ID's und Benutzerkennungen der ausgewählten Dateien angezeigt. Der Systemverwalter kann den CFN-Modus zum Standard erklären (Änderung im Modul CFSMAIN).

NCFN No Complete Filename. Standardformat für die Ausgabe von Dateinamen in CFS. Es werden die Cat-ID's und Benutzerkennungen der selektierten Dateien nur dann in der Dateienliste ausgegeben, wenn dies zur Unterscheidung von anderen Dateien notwendig ist.

Zuordnung mit *-Action Code löschen

CL* Clear Action-Code *. Alle Zuordnungen: Action-Code *[*n*] <--> Name werden gelöscht. Fortan kann das Zeichen *[*n*] in Kommandos verwendet werden, ohne daß eine ungewollte Substitution durch den Namen eines zuvor mit dem Action-Code '*[*n*]' markierten Datenobjekts erfolgt.

Beispiel:

```
/FSTAT :*:*SDF*
```

:*:*SDF* ist hier ein Parameter des FSTAT-Kommandos. Die Zeichen * sollen hier von CFS nicht durch irgendwelche Namen ersetzt werden.
Zum gleichen Thema siehe Seite **181**/**209**.

Kommandogedächtnis komprimieren

CM Compress Memory. Die verschiedenen Tabellen des Kommandogedächtnisses (FILENAME-SELECT, COMMAND, Connections) werden komprimiert, indem gleiche Einträge aussortiert werden. Die Komprimierung bewirkt eine Reorganisation und verhindert den Überlauf der Tabellen. Die chronologische Reihenfolge der Einträge in den einzelnen Tabellen geht bei der Komprimierung verloren.

Zum gleichen Thema siehe auch Kommandos AMC (Automatic Memory Compression) Seite **469**, LM/SM (Load Memory/Save Memory) Seite **252**/**281**, sowie **59**, **211**, **353**.

Vergleichen von Datenobjekten

Die Funktion COMP bietet folgende Möglichkeiten:

a) **Vergleich von zwei Datenobjekten:**
zwei Dateien mit beliebigem Dateiformat (Fcbtype, Recform, Recsize, ...),
zwei Bibliothekselemente, bzw. eine Datei mit einem Bibliothekselement.
zwei Jobvariablen.

COMP *name1* , *name2* [,SYNC=*s*] [,PRINT=<u>Y</u>|N|R|H [*w*]] [,STRIP=Y|N] [,CFIELD=*cfld*] [,MODE=<u>STD</u>|FAST] [,?]

b) **Vergleich von zwei Bibliotheken** auf der Ebene der einzelnen Elemente

COMP *bib1*[(*such*)] , *bib2* [,NAMETR=*ntr*] [,LMODE=<u>TOC</u>|SHORT|DATA] [,PRINT=<u>Y</u>|N|R|H [*w*]] [,VERS=ALL] [,MODE=<u>STD</u>|FAST]

c) **Versionsvergleich**
In einer Bibliothek werden alle Elemente verglichen, die in mehr als einer Version vorliegen.
Die Elemente werden gemäß aufsteigender Versionsnummern miteinander verglichen.

COMP *bibl1*[(*such*)] [,VERS=*nn*] [,PRINT=<u>Y</u>|N|R|H [*w*]] [,MODE=<u>STD</u>|FAST]

d) **Dateilistvergleich**
alle Dateien/Bibliotheken, deren Namen in zwei von CFS erzeugten Dateienlisten aufgeführt sind.

COMP (*name1*) , (*name2*) [,NAMETR=*ntr*] [,LMODE=<u>TOC</u>|SHORT|DATA] [,PRINT=<u>Y</u>|N|R|H [*w*]] [,VERS=ALL] [,MODE=<u>STD</u>|FAST]

Als Ergebnis des Vergleichs wird ein Protokoll mit dem Namen CFS.COMPLIST.*tsn.yymmdd.hhmmss* erzeugt. Die Ausgabe der Unterschiede ist analog dem EDT-Compare.

Der Umfang des Vergleichs kann über Parameter bestimmt werden:

- schneller Vergleich auf Gleichheit/Ungleichheit: Parameter SYNC=NO
- Synchronisierung gleicher Satzbereiche aufgrund der gesamten Satzinhalte bzw. der ISAM-Schlüssel: Parameter SYNC=RECORD|KEY.
- im Versionsvergleich (c): Alle oder nur die jeweils letzten *nn* Elementversionen: Parameter VERS=ALL | *nn*

Die Ausgabe in die Protokolldatei erfolgt je nach Parametereinstellung in unterschiedlichem Umfang.

- Ausgabe der unterschiedlichen Sätze in einem EDT-kompatiblen Format: PRINT=Y [*w*] (ggf. mit Umbruch in Teilsätze in der Länge *w*).
- Ausgabe der unterschiedlichen Sätze im character und hexadezimalen Format: PRINT=H

Alle Parameter des COMP-Kommandos sind wahlweise. Falls keine Parameter angegeben wurden, bzw. falls als letzter Parameter "?" angegeben wurde, so wird die nachfolgend dargestellte COMP-Parameter Maske mit den zuletzt gültigen Parameterwerten angezeigt.

```
 dd.mm.yy   hh:mm:ss       HOST: ........    TSN: ....      USER-ID: ........

                          Parameters for File-Compare (COMP)

 File/Library/JV 1 :
 File/Library/JV 2 :

 Synchronisation          (SYNC=Record/Key/N/col) :
 Print Options                    (PRINT=Y/N/H/R) : Y
 Strip Blanks at end of record        (STRIP=Y/N) : N
 Compare Fields                   (CFIELD=+/-col) :


 Options for Library or Filelist Compare:

 Library Mode              (LMODE=Toc/Short/Data) : TOC
 Version Compare                  (VERS=ALL/nnnn) :
 Name Transformation       (NAMETR='str1'='str2') :




 For Help:   type  "?"  at any input-field.                    Terminate with K1
```

Parameter des COMP-Kommandos:

name1 Name des ersten Datenobjekts für den Vergleich. Als Datenobjekt kann hier angegeben werden der Name einer Datei, einer Jobvariablen, einer Bibliothek bzw. eines Bibliothekselements.

Ein Bibliothekselement wird in der Form *bibl*([*t*/] *name* [/*vers*]) angegeben. Bei PLAM-Bibliotheken kann der Elementtyp S/ auch weggelassen werden.

Sollen bestimmte Elemente einer Bibliothek mit den entsprechenden Elementen einer anderen Bibliothek verglichen werden, so können die gewünschten Elemente der ersten Bibliothek durch eine Suchanweisung der folgender Form bestimmt werden: *bibl*('*str1*' [*vk* '*str2*'] ...). Als Verknüpfungsoperator *vk* kann ,|+|* (Oder/Und/ Wildcard) angegeben werden. Für eine ausführliche Beschreibung der Suchanweisung siehe Abschnitt "Mehrfachsuche", Seite 309.
Beispiel: `CFS.LIB.TEST1('S/','J/')`. Aufgrund dieser Auswahlbedingung werden alle Elemente vom Typ S oder J in der PLAM-Bibliothek CFS.LIB.TEST1 mit entsprechenden Elementen der bei *name2* angegebenen Bibliothek verglichen.

Sollen alle Versionen eines Bibliothekselements in aufsteigender Reihenfolge miteinander verglichen werden (Versionsvergleich), *name2* nicht angegeben), so darf die Versionsbezeichnung */vers* im Elementnamen nicht angegeben werden. Sollen die Versionen eines Elements ab einer bestimmten Nummer bzw. bis zu einer bestimmten Nummer miteinander verglichen werden, so muß am Ende der Versionsbezeichnung */vers* der Zusatz + bzw. - angegeben werden.
/vers+
Alle Versionen von der angegebenen Nummer bis zur höchsten vergleichen.
/vers-
Alle Versionen von der niedrigsten bis zur angegebenen Nummer vergleichen.

(*name1*) Dateilisten Vergleich: Name der ersten für den Vergleich als Referenz zu verwendenden Dateienliste. Die Dateienliste ist zu erstellen durch Selektion der gewünschten Datenobjekte und durch Sichern dieser Auswahl durch das Kommando DOC *datei*;AU (siehe 473).
Vor der Ausführung des DOC-Kommandos ist darauf zu achten, daß in der Dateienliste das Alter und nicht das Datum der Dateien angezeigt wird. Evtl. ist hierzu das Kommando AGE einzugeben.

Mit Hilfe des Dateilisten Vergleichs können mehrere Bibliotheken als Dateien ausgewählt und in einem Compare-Lauf mit den in der zweiten Referenzdatei (*name2*) enthaltenen Bibliotheken auf der logischen Ebene der einzelnen Elemente verglichen werden. Siehe hierzu auch den Parameter LMODE und VERS.

name2 Name des zweiten Datenobjekts für den Vergleich. Beim Vergleich aller Elemente mit mehreren Versionen in einer Bibliothek (Versionsvergleich darf *name2* nicht angegeben werden (COMP *bib1,,params*) bzw. es muß das entsprechende Feld in der Maske auf Blank gesetzt werden.

(*name2*) Dateilist-Compare: Name der zweiten, als Referenz für den Vergleich zu verwendenden Dateienliste.

SYNC= REC | KEY | NO | :*col1-col2*[A|D], ...:

Synchronisierung gleicher Datenbereiche.

REC Für die Synchronisierung gleicher Satzbereiche wird der gesamte Satzinhalt verwendet. Bei ISAM-Dateien wird der Schlüssel nicht berücksichtigt.
Diese Option ist der Standard bei SAM-Dateien und ISAM-Dateien mit Keypos/Keylen gleich 5/8.

KEY Diese Option ist nur von Bedeutung für ISAM-Dateien: Es wird nur der Inhalt der ISAM- Schlüssel zur Synchronisierung gleicher Datenbereiche herangezogen.
Die Option ist Standard bei ISAM-Dateien mit Keypos/Keylen ungleich 5/8.
Das Ausgabeformat des Compare verändert sich insofern, als Datensätze mit gleichem Key einander zugeordnet sind und die Unterschiede im Satz bytegenau angezeigt werden.

NO Der Vergleich wird ohne Synchronisierung durchgeführt. Die beiden Dateien werden 1 zu 1 verglichen. Das Protokoll kann hier sehr umfangreich werden, es sei denn, der Parameter PRINT=N ist angegeben. Diese Option ist der Standard bei PAM-Dateien.

:*col1-col2*[A|D], ...:
Die Synchronisationsfelder sind per Spaltenangabe definiert.
Standardmäßig wird eine aufsteigende Sortierung angenommen. Mit dem Zusatz D kann aber auch eine absteigende Sortierung festgelegt werden.
Es sind mehrere, durch Komma getrennte Synchronisationsfelder möglich.

PRINT= Y | N | R | H [*w*]

Ausgabeumfang im Vergleichsprotokoll.

Y [*w*] Gleiche Satzbereiche werden wie im EDT nur mit den entsprechenden Satznummern angezeigt. Unterschiedliche Sätze werden in ihrer vollen Länge im Protokoll aufgeführt. Zur besseren Lesbarkeit kann wahlweise nach *w* Stellen ein Umbruch der Daten vorgenommen werden ($50 < w < 32000$).

N No. Es wird kein Protokoll erzeugt. Es wird lediglich eine Meldung in der Systemzeile bzw. Compare-Maske über die Gleich-/Ungleichheit der Dateien erzeugt. Der Vergleich wird beim ersten Unterschied beendet. PRINT=N beinhaltet SYNC=NO

R Range. Es werden nur die Satznummern der gleichen und unterschiedlichen Satzbereiche im Protokoll ausgegeben.

H [*w*] Hexadezimal. Unterschiedliche Sätze werden in Character- und darunter in hexadezimaler Form aufbereitet ausgegeben. Die Sätze werden in ihrer vollen Länge im Protokoll aufgeführt. Zur besseren Lesbarkeit kann wahlweise nach *w* Stellen ein Umbruch der Daten vorgenommen werden.

STRIP= Y | N

Blanks am Satzende mit vergleichen / ausblenden.

Y Blanks am Satzende werden beim Vergleich nicht berücksichtigt.

N Datensätze werden exakt verglichen. Es macht einen Unterschied aus, wenn in der einen Datei ein sonst gleicher Satz am Ende Blanks enthält, die in der anderen Datei im entsprechenden Satz fehlen.

CFIELD= +|-:*col1*-*col2* [=*col3*] [,...] :

Compare fields. Spaltenbereiche, die beim Vergleich ein-/ausgeschlossen werden. Es werden nur die angegebenen Spalten eingelesen (Speicherersparnis bei großen Dateien).

+:*col1*-*col2* [=*col3*] [,...] :
Für den Vergleich der Datenobjekte werden nur die angegebenen Spaltenbereiche herangezogen.
=*col3*: Der Spaltenbereich :*col1*-*col2*: in der ersten Datei wird mit dem bei *col3* beginnenden Spaltenbereich in der zweiten Datei verglichen.

-:*col1*-*col2* [=*col3*] [,...] :
Die angegebenen Spaltenbereiche werden vom Vergleich ausgeschlossen.

LMODE= TOC | SHORT | DATA

Library Mode. Vergleichsmodus von Bibliotheken.

TOC Beim Vergleich von zwei Bibliotheken wird im Protokoll lediglich angezeigt, welche Elemente in einer der Bibliotheken enthalten sind und in der jeweils anderen keine Entsprechung haben. Bei gleichnamigen Elementen wird im ein unterschiedliches Änderungsdatum im Protokoll festgehalten.

SHORT Bei entsprechenden Elementen in beiden Bibliotheken wird ein schneller 1 zu 1 Vergleich durchgeführt und bei Ungleichheit eine entsprechende Zeile im Protokoll ausgegeben.

DATA Bei entsprechenden Elementen in beiden Bibliotheken wird ein Vergleich durchgeführt. Die unterschiedlichen Sätze werden im Protokoll angezeigt.

VERS= ALL | *nn*

Version Compare. Vergleich mehrerer Elementversionen in Bibliotheken.

ALL Beim Vergleich von zwei Bibliotheken werden bei entsprechenden Elementen standardmäßig nur diejenigen mit der höchsten Versionsnummer verglichen. Durch den Parameter VERS=ALL werden alle vorhandenen Elementversionen in den beiden Bibliotheken miteinander verglichen.

nn nur für Versionsvergleich der Elemente in einer Bibliothek:
Es werden nur die letzten *nn* Versionen jedes Elements miteinander verglichen. Standardmäßig wird beim Versionsvergleich über alle Elementversionen jeweils ein Vergleich mit der nächsthöheren Version durchgeführt.

NAMETR= '*str1*'='*str2*' [, ...]

Namenstransformationsregeln.

Nur von Bedeutung bei Vergleich zweier Bibliotheken bzw. Dateilist Vergleich:
Es können **Namenstransformationsregeln** angegeben werden, die bestimmen, wie aus den in der ersten Dateienliste aufgeführten Namen entsprechende Namen in der zweiten Dateienliste gesucht werden bzw. wie aus Elementnamen der ersten Bibliothek entsprechende Elementnamen in der zweiten Bibliothek gesucht werden. Im Standardfall müssen die einander entsprechenden Namen exakt gleich sein.

In den Namen der in der Dateienliste (*name1*) aufgeführten Dateien/Bibliotheken bzw. in den Namen der in der Bibliothek *name1* enthaltenen Elemente wird die Zeichenfolge '*str1*' gesucht und durch '*str2*' ausgetauscht. Der so gebildete neue Name wird in der Dateienliste (*name2*) bzw. in der Bibliothek *name2* gesucht.
Durch ''='*str2*' kann dem Namen 1 ein Prefix vorangestellt werden.
Durch '⊔'='*str2*' kann dem Namen 1 ein Suffix angehängt werden.
Falls '*str1*' im Namen 1 nicht vorkommt, so erfolgt keine Ersetzung durch '*str2*' und Name 1 wird ohne Veränderung in der Dateienliste 2 bzw. Bibliothek 2 gesucht. Es können mehrere Transformationsregeln, getrennt durch Komma, angegeben werden.
Standard: keine Namenstransformationsregeln. Es wird nach genau übereinstimmenden Namen gesucht.

MODE= STD | FAST

Vergleichsalgorithmus.

STD Es wird der Vergleichsalgorithmus des alten COMP (siehe COMP0) verwendet. Gegenüber dem EDT-Compare werden hierbei Verschiebungen von Datensätzen erkannt.

FAST Es wird ein neuer (schnellerer) Vergleichsalgorithmus angewendet. Hierbei werden (ähnlich dem EDT-Compare) Verschiebungen als Einfügungen ausgewiesen.

Bemerkungen und Hinweise:

Das Kommando COMP ist eine Neufassung der Funktion zum Vergleichen von Dateien. Aus Kompatibilitätsgründen kann mit COMP0 auch die alte Compare Funktion aufgerufen werden (Hilfefunktion: ?COMP0). Bei Angabe alter Parameter im Kommando COMP wird automatisch in das alte COMP-Modul verzweigt.

Alle Parameter des COMP-Kommandos, außer den Dateinamen sind optional. Die entsprechenden Standardwerte sind so eingestellt, daß in den meisten Fällen die Erwartungen der Benutzer getroffen sein sollten.

Beim Ausfüllen der COMP-Maske werden je nach Dateiart und Dateiformat auf intelligente Weise Standardwerte für nicht angegebene Parameter angenommen. Durch Betätigen der DUE2-Taste wird die Maske mit allen eingesetzten Standardwerten noch einmal ausgegeben.

Das Vergleichsprotokoll wird in eine Datei mit dem Namen CFS.COMPLIST.*tsn*.*yymmdd*.*hhmmss* geschrieben. Der Benutzer kann den Namen des Protokolls aber auch selbst bestimmen, indem vor dem Vergleich ein FILE-Kommando mit dem Linknamen COMPLIST abgesetzt wird (/FILE *datei*,LINK= COMPLIST).

Hat der Benutzer den Namen des Vergleichsprotokolls per File-Kommando zugewiesen, so werden alle nachfolgenden Vergleichsprotokolle in diese Datei geschrieben. Dies geschieht solange, bis ein Kommando /REL COMPLIST eingegeben wird.

Nach einem Compare wird von CFS der Prozeßschalter 31 gesetzt (SETSW ON=31), falls die Dateien/Bibliothekselemente nicht vollständig gleich waren.

Die Bezeichnungen C1 bzw. C2 für die Parameter *name1* und *name2* haben eine besondere Bedeutung und werden nicht als Dateinamen interpretiert: Diese Angaben bewirken, daß eine mit /FILE *datei*,LINK=C1/C2 zugewiesene Datei als Eingabedatei-1/2 verwendet wird.

Beispiele:

```
COMP BIBL.PLAM(-'R/'),BIBL.PLAM.ALT,LMODE=DATA
```
Es werden Elemente der Bibliothek BIBL.PLAM, die nicht vom Typ R (Relocatable) sind, mit entsprechenden Elementen der Bibliothek BIBL.PLAM.ALT verglichen. Im Vergleichsprotokoll werden die zwischen den einzelnen Elementen unterschiedlichen Sätze mit Satznummer und Inhalt aufgelistet.

```
COMP BIBL.PLAM
```
Es werden alle Elemente, die mit mindestens zwei Versionen in der Bibliothek BIBL.PLAM vertreten sind, verglichen. Dabei wird die erste Version eines Elements mit der zweiten, die zweite mit der dritten usw. verglichen. Als Ergebnis der Vergleichs wird ein Compare-Protokoll erzeugt, in dem die Änderungshistorie einer jeden Version zur nächsten dokumentiert ist.

```
COMP (DOC1),(DOC2),LMODE=TOC,NAMETR=' '='.NEU'
```
DOC1 ist eine mit dem Kommando DOC erstellte Dateienliste von CFS, die die Namen von vier Makrobibliotheken enthält: SIPLIB.BS2CP.180, SYSLIB.BS2CP.180, SIPLIB.BCAM.200, SYSLIB.BCAM.210. DOC2 ist eine entsprechende Datei, die die Namen der Makrobibliotheken in einer neueren Fassung enthält: SIPLIB.BS2CP.180.NEU, SYSLIB.BS2CP.180.NEU, SIPLIB. BCAM.200.NEU, SYSLIB.BCAM.210.NEU.

Durch das COMP-Kommando werden die Elementnamen der jeweils zusammengehörigen Bibliotheken gegeneinander abgeglichen. Dies erfolgt in der Weise, daß Elementnamen gesucht und entsprechend gekennzeichnet werden, die nur in der einen oder anderen Bibliothek enthalten sind. Bei entsprechenden Elementen in beiden Bibliotheken wird das Änderungsdatum der Elemente verglichen und das ältere bzw. neuere Element im Vergleichsprotokoll entsprechend gekennzeichnet.
Im Unterschied zu LMODE=DATA erfolgt kein Vergleich der Elementinhalte.

Bildschirmmasken vor der Ausgabe komprimieren

CSO [STAT] | NCSO

Compress Screen Output on/off. Von CFS ausgegebene Bildschirmmasken und Formatausgaben in Connections werden vor der Darstellung am Bildschirm, d.h. bevor sie über die Leitung gesendet werden komprimiert. Die Komprimierung besteht darin, daß aufeinanderfolgende gleiche Zeichen durch spezielle Steuerzeichen (WDH-Folgen) ersetzt werden. Um optimale Komprimierungsraten zu erzielen, wird die Dateienliste von CFS in einem leicht modifizieren Format dargestellt.

Der CSO-Modus hat im Datei Display noch einen Nebeneffekt:
Im Non Modify Modus werden die durch ein Suche-Kommando gefundenen Treffer am Bildschirm hell hervorgehoben dargestellt. Durch Betätigen der F4-Taste kann dieses Highlighting auch ohne direkt vorausgehendes Suche-Kommando angefordert werden.
Im Modify-Modus ist das Highlighting Feature nicht aktiv.

STAT — Es wird der momentane Stand der Kompressionsstatistik in des Systemzeile des Bildschirms ausgegeben.

NCSO — No Compress of Screen Output. Die Bildschirmkomprimierung wird ausgeschaltet.
Das Treffer-Highlighting von Suchitems im Display-Modus ist nicht aktiv.
Standard: CSO.

Bibliothek anlegen

CREALIB *bibliothek* [, P|L|F] [, N]

Falls eine Bibliothek mit dem angegebenen Namen noch nicht existiert, wird sie als leere Bibliothek neu angelegt. Das Kommando ist besonders in CFS-Prozeduren von Nutzen, da hiermit sichergestellt werden kann, daß eine Bibliothek mit einem bestimmten Namen und Typ tatsächlich existiert. CFS stellt dann keine (in der Prozedur zu beantwortende) Zwischenanfrage, ob eine Bibliothek angelegt werden soll.

P|L|F — Typ der Bibliothek.
P PLAM
L LMS-ISAM
F FMS
Standard: P

N — New. Die Bibliothek wird in jedem Fall als leere Bibliothek angelegt. Gegebenenfalls wird eine existierende Bibliothek vorher gelöscht.
Standard: Eine bereits bestehende Bibliothek wird unverändert beibehalten.

Überprüfung des Elementtyps bei Aufnahme von Elementen in PLAM-Bibliotheken

CT | CTQ | NCT

Check Types/Check Types and Query/No Check of Types. Es erfolgt eine Überprüfung des Elementtyps bei Aufnahme von Dateien in PLAM-Bibliotheken.

CT — Es sind nur die LMS-kompatiblen Elementtypen C/D/H/J/L/M/P/R/S/X erlaubt.

CTQ — Bei Angabe nicht LMS-kompatibler Bibliothekstypen wird der Benutzer gefragt, ob er eine Aufnahme unter dem angegebenen Typ wirklich wünscht.

NCT — Keine Typ-Überprüfung bei Aufnahme von Dateien in PLAM-Bibliotheken.
Standard: CTQ.

Age-Angabe in Datumsform

DATE

Altersangaben, insbesondere die Angaben in der AGE-Spalte der Dateienliste, werden nicht als Anzahl von Tagen, sondern in Form eines Datums (dd.mm.yy) angezeigt. In der User Option-Spalte kann das Datum aus Platzgründen nur in der Form YMMDD angezeigt werden. Y ist hierbei die letzte Stelle des Jahres (z.B. 6 für 1996). Durch das Kommando NDATE [oder AGE] kann dieser Modus wieder auf die Standardeinstellung zurückgesetzt werden.
Standard: AGE.

Bibliothekselement löschen

DEL [*t/*] *element* [, *bibliothek*]

Element aus der angegebenen Bibliothek löschen.
Die Angabe *bibliothek* kann entfallen, falls ein Element aus der im letzten ADD-/DEL-/SEL-Kommando angesprochenen Bibliothek gelöscht werden soll.

Hinweis:

Die Angabe des Bibliotheksnamens in den Kommandos ADD/SEL/ DEL kann entfallen, wenn die zuletzt angesprochene Bibliothek auch im aktuellen Kommando wieder verwendet wird.

CFS-Prozedur ausführen

DO [N,] [*datei* | *element* | *bibl*(*element*) | %*name* | %?] [, *params*, ...]

Es wird die angegebene CFS-Prozedur zur Ausführung gebracht. Für eine ausführliche Beschreibung des DO-Kommandos siehe Seite 530.

Hinweis:

Das Kommando DO wird von CFS in folgender Weise abgearbeitet: Falls *bibl* (*element*) bzw. %*name* angegeben wurde, so ist dadurch die auszuführende CFS-Prozedur eindeutig bestimmt. In allen anderen Fällen wird gemäß der folgenden Hierarchie nach der Prozedur gesucht:

a) CFS sucht in der zugewiesenen Userlib (Standard: CFS.USERLIB) nach einem Element mit dem angegebenen Namen.

b) Es wird in der zentralen USERLIB (/FILE ...,LINK=USERLIB bzw. Bibliothek mit dem Namen $*user-id*.CFS.USERLIB.STD, $*user-id*=Benutzerkennung in der die Programmphase CFS gespeichert ist) nach einem Element mit dem angegebenen Namen gesucht.

c) Es wird eine Datei mit der angegebenen Bezeichnung gesucht.

Wenn a), b) und c) nicht zu einem Erfolg geführt haben, wird eine Fehlermeldung ausgegeben. Zum Verändern der Standard CFS-Userlib siehe Kommando UL, Seite 483.

Dokumentationsdatei definieren

DOC Feld DOCUMENTATION der Selektionsmaske anzeigen.

DOC *datei* Feld DOCUMENTATION direkt mit dem angegebenen Inhalt belegen.
Damit die DOC-Datei sofort erstellt wird, sollte das Kommando AU angegeben werden. Dies kann auch in Kombination mit dem DOC-Kommando erfolgen:
DOC *datei*;AU

Für weitere Informationen wird auf die Beschreibung des Feldes DOCUMENTATION Seite 74 verwiesen.

DO-Parameter Maske bei Prozeduren ausgeben

DP | NDP Do-Parameter mask/No Do-Parameter mask.

DP Bei einem BS2000 DO-Kommando, das im Kommandofeld der Dateienliste eingegeben wurde bzw. bei DO-Kommandos in Connections, wird die PROC-Anweisung der DO-Prozedur analysiert. Wird dabei festgestellt, daß DO-Parameter existieren, die noch nicht mit einem Wert versorgt sind, so wird dem Benutzer eine Parameter-Maske angeboten. In ihr können die noch fehlenden DO-Parameter nachträglich angegeben werden.
Standard: DP.

Hinweis:

Bei DO-Kommandos in Connections wird die Parameter-Maske auch im DP-Modus nicht ausgegeben, falls der Nebenprozeß unter einer anderen Benutzerkennung als der Grundprozeß von CFS läuft und die angesprochene DO-Prozedur nicht shareable ist bzw. falls die Connection auf einem anderen als dem eigenen Host-Rechner eröffnet wurde.

Display used resources

DU [M | CPU | WSI | IO]

In der Maske der Dateienliste wird anstelle der Uhrzeit (*hh*:*mm*:*ss*), eine Angabe zu den im letzten Transaktionsschritt verbrauchten Systemresourcen gemacht.

M Display Used Memory. Es wird die Größe des belegten Klasse-6 Speichers im Adressbereich unterhalb 16MB in K (Kilobyte) angezeigt.

CPU Display Used CPU-Time. Es wird die seit der letzten Bildschirmausgabe verbrauchte CPU-Zeit angezeigt.

WSI Display Used Working Set Integral. Es wird der Wert des Working Set Integrals seit der letzten Bildschirmausgabe angezeigt. Das Working Set Integral ist definiert als das Produkt aus Hauptspeicherseiten * verbrauchte Zeit.

IO Display Used IO's. Es wird die Anzahl der seit der letzten Bildschirmausgabe angefallenen Platten-IO's angezeigt.

NDU Anstelle der verbrauchten Systemresourcen wird die Uhrzeit (*hh*:*mm*:*ss*) angezeigt.
Standard: NDU.

DO-Prozedur/Enter-Datei über das CFS Job-Report System starten

E *proc* [, *do-params*] [, ?] [/*enter-params*] [, JRF=*jr-file*] [, ?] [, JN=*jobname*]

Starten und Überwachen eines Enter-Job's. Es kann auch eine DO-Prozedur als Enter-Job gestartet werden. In diesem Fall werden die DO-Parameter analysiert und formatgesteuert angefordert. Bei Jobende wird dem Benutzer das Ergebnis in Form des SYSOUT-/SYSLST-Protokolls oder des Inhalts einer beliebigen JR-File angezeigt.
Die Enter-Parameter können auch mit dem JRP-Kommando vorbelegt werden.
Weitere Informationen zu diesem Thema finden Sie auf Seite 397 und folgende (Job-Report System).

Aufruf des EDT als Unterprogramm

EDT[*n*] [, *datei*[,O] | *bibl*([*t*/] *element* [/*vers*]) | * | D]

In den F-Modus des EDT verzweigen (Rückkehr durch H [ALT] oder K1-Taste).

n Der Datenbereich der angegebenen EDT-Ebene wird aktiviert.
Standard: *n* = 0.

datei[,O] In den Datenbereich des EDT wird die angegebene Datei geladen.
Der Zusatz O bewirkt, daß die Datei mit der EDT-Anweisung OPEN und nicht mit READ/GET eingelesen wird. Das Zurückschreiben der Datei erfolgt in diesem Fall mit CLOSE und nicht mit WRITE/SAVE.

bibl([*t*/] *element* [/*vers*])
Es wird das angegebene PLAM-Element in den Datenbereich des EDT geladen (@OPEN-Anweisung). Das Bibliothekselement bleibt bis zum Zurückschreiben mit der Anweisung @CLOSE gegen Änderungen durch andere Benutzer gesperrt.
Deltagespeicherte PLAM-Elemente können nicht bearbeitet werden.

* In den Datenbereich des EDT wird die mit dem Action-Code * markierte Datei/ das markierte Bibliothekselement geladen.
PLAM-Bibliothekselemente bleiben bis zum Zurückschreiben mit der Anweisung @CLOSE gegen Änderungen durch andere Benutzer gesperrt.

D In den Datenbereich des EDT wird die im CFS-Display angezeigte Datei geladen.

Hinweise:

Für das Zurückschreiben der im EDT bearbeiteten Datei muß der Benutzer selbst sorgen (EDT-Anweisungen @W (SAM-Datei), @SA (ISAM-Datei) bzw. @CL (PLAM-Element)). Ein @FILE-Kommando mit dem Namen der geladenen Datei wird von CFS abgesetzt.

Bei EDT,* / EDL,* für FMS- und LMS-ISAM Bibliothekselemente wird nach dem Zurückschreiben der Hilfsdatei im EDT mit @WRITE und nach der Rückkehr aus dem EDT mit ADD * automatisch ein Update des Elements angeboten.

Einmal eingestellte EDT-Parameter, wie z.B. LOWER ON, SCALE, usw. bleiben über die gesamte CFS-Sitzung erhalten. Das gleiche gilt für die Inhalte aller EDT-Ebenen, sowie der Zeilen- und Spaltenpositionen in den einzelnen EDT-Ebenen.

Non-Standard Dateien, z.B. mit RECFORM=F oder ISAM-Dateien mit KEYPOS/ KEYLEN ungleich 5/8 können einfach bearbeitet werden. Eine manuelle Zuweisung über die EDT-Linknamen EDTSAM/EDTISAM ist nicht notwendig. Diese wird von CFS selbst vorgenommen.

Vor Verlassen des EDT mit K1 müssen zuerst alle noch anstehenden Änderungen mit ENTER ausgeführt werden. Die Taste **K1** beendet den EDT ohne Berücksichtigung von evtl. im gleichen Bildschirm vorgenommenen Datenänderungen.

Bei eingeschaltetem QR-Modus (Query on Return) wird der Benutzer nach der Rückkehr aus dem EDT gefragt, ob der virtuelle EDT-Speicher gelöscht werden soll (@D). Der QR-Modus ist standardmäßig eingeschaltet und kann mit dem Kommando NQR ausgeschaltet werden.

Über das Kommando EDT wird der normale EDT im Format-Modus angesprochen. Falls neben EDT auch der TOM-TI Editor zur Verfügung steht, so kann der System-Administrator CFS so generieren, daß mit EDT/EDF anstelle des normalen EDT der TOM-TI Editor angesprochen wird. In diesem Falle stehen dem Benutzer eine Reihe zusätzlicher TOM-TI-Kommandos zur Verfügung (z.B. SORT/ COL/ REF/ DOC/ FETCH).

Im Formatmodus des EDT gelangt der Benutzer entweder durch das Kommando HALT oder durch Betätigen der K1-Taste in das Programm CFS zurück. Im HALT-Kommando des EDT kann der Benutzer einen optionalen Zusatztext angeben (HALT *cmd*). Dieser Zusatztext wird von CFS als Kommando interpretiert und ausgeführt.

Die Kommandos EDT/EDF/EDL und die gleichlautenden Action-Codes EDT/ EDF/ EDL rufen zwei in ihren Datenbereichen getrennte EDT's auf.
Die Kommandos EDT/EDF/EDL benutzen den gleichen EDT-Datenbereich.

Unter CFS ist der INTR-STXIT des EDT nicht aktiviert, d.h. das BS2000-Kommando INTR wirkt genauso wie das Kommando R [esume]. Hierin besteht ein Unterschied zum Aufruf des EDT als eigenständiges Programm: /EXEC $EDT.

Falls die Job-Report Funktion von CFS die EDT-Sitzung des Benutzers unterbricht, so kann nach Beendigung des Job-Report (K1-Taste, END, N) der ursprüngliche EDT-Bildschirm mit der K3-Taste wieder hergestellt werden.

Automatische Kommandos nach Aufruf des EDT

EDTINIT [*cmd;cmd; ...*]

Nach dem Einlesen eines Datenobjekts mit dem Action-Code EDT[*n*] oder dem Kommando EDT, werden die angegebenen EDT-Kommandos ausgeführt.

cmd Ein oder mehrere EDT-Kommandos. Mehrere Kommandos sind mit dem Zeichen ';' zu trennen. Beispiel: `EDTINIT LOWER ON;PAR EDIT-FULL=ON`
Standard: Es werden keine benutzerdefinierten EDT-Kommandos ausgeführt.

Hinweise:

Beim gleichzeitigen Einlesen mehrerer Datenobjekte in verschiedene Ebenen des EDT (Action-Code EDT[*n*]) werden die EDTINIT-Kommandos in jeder EDT-Ebene einzeln ausgeführt.

Das Kommando EDTINIT ohne Parameter bewirkt, daß alle zuvor definierten Initialisierungskommandos zurückgesetzt werden.

Beispiel:

```
EDTINIT INF=ON,LOWER ON;PAR EDIT-FULL=ON
```

Verarbeitungsmodi des EDT bestimmen

EDTMODE [UNI|COMP] [OPENSTD=Y|N]

UNI | COMP Nur relevant ab EDT V17.0
Mit diesen Angaben kann der Benutzer festlegen, ob EDT in einem Modus ablaufen soll, der kompatibel ist zu EDT V16.6 (COMP), oder ob der EDT die neue Unicode-Funktionalität nutzen soll (UNI). Letzteres beinhaltet auch die Möglichkeit, Dateien mit beliebig langen Sätzen zu bearbeiten.
Der Standardwert ist UNI, falls EDT > V16.6 installiert ist.

Nachdem der EDT im Kommando oder als Action-Code geladen wurde, ist eine erneute Änderung des Modus nicht möglich. Jedoch kann der Modus zwischen Kommando und Action-Code EDT verschieden eingestellt werden.
Beispiel:
EDTMODE COMP
EDT (EDT im Kommandofeld im kompatiblem Modus betreiben)
H (EDT verlassen)
EDTMODE UNI
EDT (als Action-Code bei einer Datei)
H

OPENSTD=Y Nur von Bedeutung für das Verhalten des Kommando EDT beim Einlesen von Dateien.
Dateien werden im EDT mit der Anweisung OPEN FILE=..., TYPE=CATALOG eingelesen und sind dadurch während der gesamten Bearbeitung für andere Benutzer gesperrt. Die Dateien müssen vor Verlassen des EDT mit der Anweisung CLOSE zurückgeschrieben und damit entsperrt werden.
Standard: OPENSTD=N Dateien werden mit den Anweisungen READ/GET eingelesen und mit WRITE/SAVE zurückgeschrieben. Die Dateien sind während der Bearbeitung nicht gesperrt.

Aufruf des EDT als Unterprogramm (Line-Modus)

EDL [, *datei* | *bibl*([*t*/] *element* [/*vers*]) | * | D]

Wie EDT (siehe oben), jedoch wird der EDT im Line-Modus aufgerufen.
Rückkehr durch @H oder @RET.

Aufruf von EDOR als Unterprogramm

EDOR [, *datei*] | EDOR , * | EDOR , D

Wie EDT (siehe oben), jedoch wird EDOR aufgerufen. Die Rückkehr von EDOR nach CFS ist möglich durch die EDOR-Kommandofolge H ! H. Die Bearbeitung der Datei wird damit beendet. Eine zweite Möglichkeit, aus dem Programm EDOR nach CFS zu gelangen, bietet die K2-Taste. Der Benutzer kann zu jedem Zeitpunkt im EDOR die K2-Taste betätigen. Er gelangt dadurch direkt nach CFS. Beim nächsten EDOR-Aufruf erhält der Benutzer den Zustand des EDOR, in dem dieser zuletzt durch Betätigung der **K2**-Taste verlassen wurde.

Erase with Retain of Tempfiles

ERT | NERT — Erase Retain Tempfiles/Erase with No Retain of Tempfiles.

ERT — Der ERT-Modus hat zur Folge, daß die mit dem Action-Code E gelöschten Datenobjekte zunächst in temporäre Dateien umkatalogisiert bzw. in temporäre Bibliotheken übertragen werden. Diese temporären Dateien werden vom BS2000 bei Prozeßende gelöscht. Versehentlich vom Benutzer mit dem Action-Code E markierte und gelöschte Daten sind damit noch bis Prozeßende physisch als Dateien vorhanden (zugreifbar durch Kommando NP# bzw. NP STAJV #).

Die Namen der erzeugten Temporärdateien werden gebildet, indem beim CAT-Kommando den Namen der zu löschenden Dateien das Tempfile-Zeichen (normalerweise #) vorangestellt wird. Daraus ergibt sich, daß der ERT-Modus beim expliziten Löschen von temporären Dateien unwirksam ist.

Bibliothekselemente werden vor dem Löschen unter dem gleichen Namen in eine temporäre Bibliothek kopiert und sind dort bis Prozeßende noch zugreifbar. Der Name der temporären Bibliothek wird gebildet, indem dem Namen der Bibliothek aus der Elemente gelöscht werden, das Tempfile-Zeichen # vorangestellt wird.

Die ERT-Option **wirkt nicht**, falls

- Dateien mit den Action-Codes ED (Erase...,Data), EC (Erase..., Catalog) oder EN (Erase No Retain Tempfiles) gelöscht werden,
- Dateien auf Privatplatten mit mehr als 5000 beschriebenen Seiten mit Action-Code E gelöscht werden,
- Dateien mit einer PVS-Bezeichnung im Feld USER-ID selektiert wurden und diese mit Action-Code E gelöscht werden,
- unter TSOS Dateien einer fremden Kennung selektiert wurden und diese mit Action-Code E gelöscht werden,
- PLAM-Bibliothekselemente mit dem Action-Code EA (Erase All delta versions) gelöscht werden.

Gelöschte Datenobjekte werden automatisch aus der aktuellen Dateienliste entfernt (impliziter Action-Code '-'). Mit dem Kommando YANK können diese unsichtbaren Zeilen der Dateienliste wieder sichtbar gemacht werden.

Die ERT-Option ist standardmäßig eingeschaltet.

NERT Die ERT-Option läßt sich ausschalten mit dem Kommando NERT (No Erase Retain Tempfiles) bzw. über die SET-Maske.

Dateiumsetzung: FC (File-Conversion)

FC [*datei1*] [, *datei2*] [, ?] [, *reclim*] [, *konst*] [, *opt*] [, LEN=*length*]

Die Eingabedatei (*datei1*) wird in eine zweite Datei (*datei2*) mit frei wählbarem Datenformat umgesetzt.

datei1 Name der Eingabedatei bzw. Komma, falls in die Ausgabedatei Sätze mit einen fest vorgegebenen Füllzeichen geschrieben werden sollen.

datei2 Name der Ausgabedatei.

Die Dateiattribute der Ausgabedatei können in der FC-Maske (siehe unten) eingetragen oder über ein zuvor abgesetztes FILE-Kommando mit LINK=OUTFCB bestimmt werden.

Die FC-Maske wird angezeigt bei Angabe des Kommandos FC ohne Parameter bzw. bei Angabe der Dateinamen, gefolgt von einem Fragezeichen (FC [*datei1*],*datei2*,?).

Das FILE-Kommando zum Festlegen der Dateiattribute der Ausgabedatei kann z.B. wie folgt aussehen: /FILE *datei2*,LINK=OUTFCB,FCBTYPE=ISAM,RECFORM=F, RECSIZE=4000,KEYPOS=20,KEYLEN=12,BLKSIZE=(STD,2),SPACE=(30,30).
Anstelle des Namens der Ausgabedatei kann im FC-Kommando auch OUTFCB angegeben werden. In diesem Fall wird der Dateiname über den Link-Namen OUTFCB bestimmt.

? Das Fragezeichen kann im Anschluß an den Namen der Ausgabedatei angegeben werden und bewirkt, daß die FC-Maske angezeigt wird. In dieser Maske können die Dateiattribute und alle im folgenden beschriebenen Parameter eingegeben werden.

reclim maximale Anzahl der Sätze, die aus *datei1* nach *datei2* zu übertragen sind.
Falls *datei1* nicht angegeben ist, werden entsprechend viele Leersätze in der Ausgabedatei angelegt.
Der Wert *reclim*=0 bewirkt, daß nur ein Open- und Close-Aufruf für die Ausgabedatei durchgeführt wird. Es werden keine Datensätze geschrieben.
Standard: *reclim*=Anzahl der Eingabesätze bzw. *reclim*=1, falls keine Eingabedatei angegeben wurde.

konst C'*string*' | X'*string*'
Gibt ein Muster vor, mit dem die Ausgabesätze aufgefüllt werden, falls diese eine größere Länge haben als die Sätze der Eingabedatei. Das Muster kann auch aus mehr als einem Zeichen bestehen.
Standard: C'␣'.

opt IK | EK | NA

IK Insert Keys: Für die Sätze der Ausgabedatei werden von CFS selbständig ISAM-Schlüssel generiert. Diese Schlüsselfelder können mit der Option IK auch bei Ausgabedateien vom Fcbtype SAM erzeugt werden. In diesem Fall erhalten die Ausgabesätze nach dem Satzlängenfeld eine 8-stellige Nummerierung.
Standard: kein IK. Ein ISAM-Schlüssel ist in den Daten der zu konvertierenden Datei enthalten. Falls diese ISAM-Schlüssel nicht in aufsteigender Reihenfolge vorliegen, ist die Option NA (Non Ascending Keys, siehe unten) anzugeben.

EK Eliminate Keys: Bei den Ausgabesätzen werden die in der Eingabedatei vorhandenen ISAM-Schlüssel ausgeblendet. Falls die Eingabedatei keine ISAM-Datei ist, bleibt die Option EK ohne Wirkung.
Standard: kein EK, d.h. bei Eingabedateien vom Typ ISAM werden die Keys beim Konvertieren nicht ausgeblendet.

NA Non Ascending Keys: Bei Konvertierung in eine ISAM-Datei enthalten die Eingabesätze die Daten für die ISAM-Schlüssel. Diese liegen jedoch nicht in aufsteigender Reihenfolge vor, so daß CFS mit dem ISAM-Makro STORE anstelle von PUT arbeiten muß. Die Angabe NA bewirkt i.a. eine Umsortierung der Sätze der Eingabedatei.
Standard: kein NA. Eine nicht aufsteigende Reihenfolge des Schlüsselbegriffs in der Eingabedatei führt auf einen DVS-Fehler.

LEN=*length* Anzahl der Bytes in den zu erzeugenden Ausgabesätzen. Ein eventueller ISAM-Schlüssel und das Satzlängenfeld sind bei der Längenangabe zu berücksichtigen. Die Angabe LEN= bewirkt, daß Recform-V Sätze in angegebenen der Länge in die Ausgabedatei geschrieben werden.
Standard: Bei Ausgabedateien mit RECFORM=V ergibt sich die Satzlänge aus der Länge der Eingabesätze.

Format der FC-Maske zur Eingabe der Dateiattribute und Parameter:

```
dd.mm.yy   hh:mm:ss       HOST: ........        TSN: ....  USER-ID: ........

                      Parameters for File-Conversion (FC)


Input- File                       :
Output-File                       :
       Fcbtype   (SAM/ISAM/PAM) : ISAM
       Recform            (F/V) : V
       Recsize                  : 0
       Blksize                  : (STD,01)
       Keypos                   : 5
       Keylen                   : 8
       Pad                      : 15
       Blkctrl (PAMKEY/DATA/NO) : PAMKEY


Options for File-Conversion
       Record-Limit             :
       Fill-Pattern (X'.'/C'.') : C' '
       Key-Option    (EK/IK/NA) :
       Output-Length   (LEN=..) :

for Help:  type "?" in any Input-Field
```

Hinweise:

Bei Programmphasen und PLAM-Bibliotheken wird die Konvertierung nach BLKCTRL=NO an das Dienstprogramm PAMCONV zur Ausführung weiter gereicht.

Falls die Ausgabedatei im Format RECFORM=U gewünscht wird, muß ebenfalls der Parameter RECSIZE=8 angegeben werden (Mehrzweckregister 8 wird im Programm für die Länge der Ausgabesätze verwendet).

Falls der Name der Eingabedatei in der FC-Maske leer gelassen wird, so wird eine Ausgabedatei im gewünschten Format mit Fill-Character aufgefüllten Sätzen erstellt. Zur Bestimmung der Satzlänge wird die Recsize-Angabe bzw. bei Recform=V die An-

gabe LEN= herangezogen. Die Anzahl der zu schreibenden Datensätze wird im Feld Record-Limit bestimmt.

Mit dem Kommando FC können auch SAM-/ISAM-Dateien in PAM-Dateien umgesetzt werden und umgekehrt. Es sind folgende Fälle zu unterscheiden:

SAM/ISAM --> PAM

1) Eingabedatei: BLKSIZE = (STD,1)
Die Eingabedatei (SAM/ISAM) wird blockweise gelesen und als PAM-Datei weggeschrieben. Es werden keine Dateninhalte verändert.

2) Eingabedatei: BLKSIZE > (STD,1)
Es wird unterstellt, daß die ersten 16 Byte der Eingabesätze der SAM-/ ISAM-Datei (Recform=V) die Daten des PAM-Keys enthalten. Daran anschließend werden 2048 Byte Daten für einen vollen PAM-Block erwartet.

PAM --> SAM/ISAM

1) Ausgabedatei: BLKSIZE = (STD,1)
Die Eingabedatei wird blockweise in die Ausgabedatei übertragen. Es werden keine Dateninhalte verändert. Es wird lediglich der FCBTYP der Ausgabedatei von PAM nach SAM/ISAM umgewandelt.

2) Ausgabedatei: BLKSIZE > (STD,1)
In den ersten 16 Datenbytes jedes Ausgabesatzes wird der PAM-Key des gelesenen Blocks abgelegt. Daran anschließend werden die 2048 Byte Daten des gelesenen PAM-Blocks angefügt.

CFS-Informationskatalog konvertieren

a) ISAM-Katalog --> SAM-Datei

FC2 EXP: *katalog*, *datei2*

Übertragen (Exportieren) der in CFS mit den Action-Codes IM/IA erzeugten Dokumentationstexte in eine Standard EDT-Datei.

katalog Name des zu exportierenden Informationskatalogs (CFS.INF bzw. *bibl*.INF).

datei2 Name der SAM-Datei, in die der Datenkatalog exportiert werden soll.

b) SAM-Datei --> ISAM-Katalog

FC2 IMP: *datei1*, *katalog* [, MERGE]

Umsetzen (Importieren) einer SAM-Datei mit den Informationen für den CFS-Datenkatalog in das von CFS benutzte ISAM-Format für Dokumentationstexte.

datei1 Name der SAM-Datei mit den Informationstexten.

katalog Name des zu erzeugenden Datenkatalogs (CFS.INF bzw. *bibl*.INF).

MERGE Die in der SAM-Datei enthaltenen Daten werden in einen bereits bestehenden Datenkatalog eingemischt.
Standard: Der Datenkatalog wird neu erstellt.

c) ISAM-Katalog --> PLAM-Bibliothek

FC2 CON: *datei1* [, *datei2*] [, P]

datei1 CFS-Informationskatalog (ISAM-Datei).

datei2 PLAM-Bibliothek (Ausgabe).
Falls *datei2* nicht angegeben wurde, so wird eine temporäre Bibliothek erzeugt, die am Ende den Informationskatalog *datei1* überschreibt.

Der Inhalt des Informationskatalogs wird in eine bereits existierende oder neu anzulegende PLAM-Bibliothek *datei2* übertragen. Für jeweils eine Dateibeschreibung wird ein Element in der Bibliothek *datei2* erzeugt: S/*xxxx*. Falls sich die Dateibeschreibung über mehrere Bildschirmseiten erstreckt (mehrere ISAM-Datensätze), so werden die entsprechenden Daten im erzeugten Bibliothekselement durch Sätze mit dem Inhalt $NEW PAGE getrennt.

d) PLAM-Bibliothek --> ISAM-Katalog

FC2 CON: *datei1* [, *datei2*] [, P]

datei1 PLAM-Bibliothek.

datei2 CFS-Informationskatalog (Ausgabe).
Falls *datei2* nicht angegeben wurde, so wird ein temporärer Informationskatalog erzeugt, der am Ende die PLAM-Bibliothek *datei1* überschreibt.

Der Inhalt der Elemente der PLAM-Bibliothek *datei1* wird in einen bestehenden bzw. neu anzulegenden Informationskatalog *datei2* übertragen. Zu einem Bibliothekselement *t/xxxxx* werden ein oder mehrere ISAM-Sätze mit Key *xxxxx* im Informationskatalog *datei2* erzeugt. Mehrere Sätze werden geschrieben, falls in den Daten des PLAM-Elements die Trennsätze $NEW PAGE enthalten sind.

P Dieser optionale Zusatz ist nur bei Konvertierung von PLAM nach ISAM von Bedeutung. Die Typbezeichnungen der Elemente in der PLAM-Bibliothek werden in die ISAM-Keys der Sätze des Informationskatalogs mit übernommen. Der so erzeugte INF-Katalog kann ausschließlich als Dokumentation für Elemente einer PLAM-Bibliothek verwendet werden.

Datei für Transfer über Internet vorbereiten

FILEXP *datei1* [, *datei2*] [, PAM]

File Export. Die Eingabedatei (*datei1*) wird in ein transparentes Datenformat umgewandelt. Die erzeugte Ausgabedatei (Fcbtype=SAM, Recform=V) besteht aus Datensätzen < 256 Byte, die nur abdruckbare Buchstaben und Ziffern enthalten. Die so erzeugte Datei kann im Text-Modus mit beliebigen File-Transfer Produkten auf andere Rechnerplattformen übertragen und von dort über das Internet mit email oder FTP als Textdatei weiter verschickt werden.

datei1 Name der Eingabedatei.

datei2 Name der Ausgabedatei (Export-Datei).
Falls nicht angegeben, wird der Standardname CFS.EXPORT.FILE verwendet.

PAM nur relevant bei der Konvertierung von SAM- oder ISAM-Dateien. Die Eingabedatei wird nicht logisch, d.h. auf Satzebene sondern PAM-Blockweise gelesen. In der Ausgabedatei werden die Abbilder der PAM-Blöcke abgelegt. Auf diese Weise können auch defekte Dateien für einen Versand exportiert werden.

Hinweise:

Die Export-Datei ist ungefähr doppelt so groß wie die Originaldatei. Nach dem Transferieren der Datei auf ein PC-System (Windows/NT) ist es empfehlenswert, die PC-Datei mit einem Packprogramm (z.B. PKZip) in eine selbstextrahierende EXE-Datei zu komprimieren. Es werden Komprimierungsraten von 25% oder weniger erreicht. Die komprimierte Datei kann mit email oder FTP im Binärmodus an gleichartige Partnerrechner (Windows/NT) übertragen werden.

Die Export-Datei enthält alle Informationen, die zur Rekonstruktion der Originaldatei benötigt werden, z.B. die originalen Dateiattribute sowie ggf. den Inhalt der PAM-Keys.

Aus Export-Datei BS2000-Datei im Originalformat erstellen

FILIMP *datei1* [, *datei2*]

File Import. Aus der Eingabedatei (*datei1*) im transparenten Datenformat wird eine BS2000-Datei im originalen Datenformat und mit dem originalen Dateninhalt rekonstruiert.

datei1 Name der Eingabedatei.

datei2 Name der Ausgabedatei (Datei im Originalformat).
Falls nicht angegeben, wird der im ersten Datensatz der Eingabedatei hinterlegte Name der Originaldatei verwendet. Eine evtl. vorhandene PVS- und User-ID wird dabei nicht berücksichtigt.

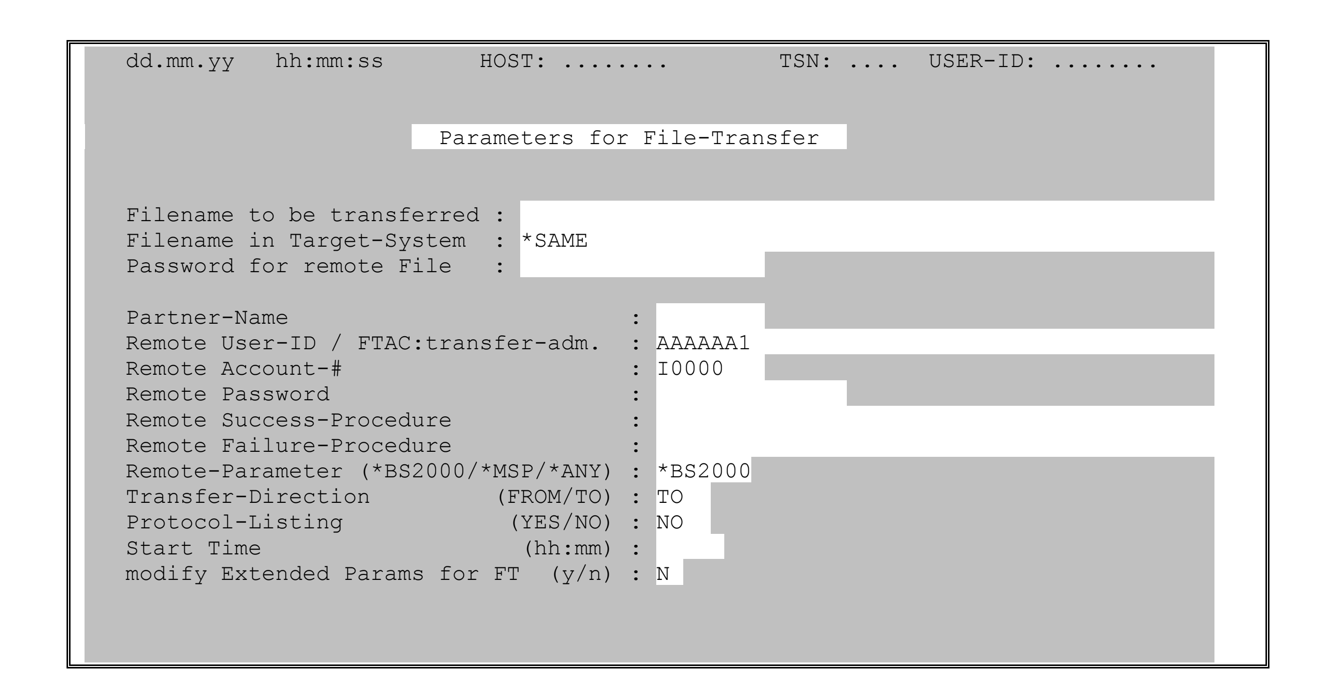
Hinweis:

Das Programm kann auch ohne CFS als Stand-Alone Programm aufgerufen werden:
/EXEC (FILIMP,$user-id.CFSLIB)
Die Ein- und Ausgabedateinamen werden in diesem Fall über SYSDTA eingelesen.

File-Transfer mit openFT

FT [%*name* | %? [, JRS=*jrfile* | ,?] [,]] [*datei1*] [, *datei2*] [, *passw*] [, *host2*] [, *user-id2*] [, *acct-nr2*] [, *passw2*] [, *rem-succ*] [, *rem-fail*] [, *rem-par*] [, *direct*] [, *lst*] [, *time*]

Alle Parameter des FT-Kommandos sind wahlweise. Falls keine Parameter angegeben sind, so werden diese in der FT-Parameter Maske (siehe unten) angefordert.



```
dd.mm.yy   hh:mm:ss      HOST: ........      TSN: ....  USER-ID: ........

                       Parameters for File-Transfer

Filename to be transferred :
Filename in Target-System  : *SAME
Password for remote File   :

Partner-Name                            :
Remote User-ID / FTAC:transfer-adm.     : AAAAAA1
Remote Account-#                        : I0000
Remote Password                         :
Remote Success-Procedure                :
Remote Failure-Procedure                :
Remote-Parameter (*BS2000/*MSP/*ANY)    : *BS2000
Transfer-Direction            (FROM/TO) : TO
Protocol-Listing               (YES/NO) : NO
Start Time                      (hh:mm) :
modify Extended Params for FT     (y/n) : N
```

Parameter des FT-Kommandos:

%name Die Parameter zum File-Transfer werden dem angegebenen Parameterset entnommen.

%? Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

JRS=*jrfile* Name der JRSAVE-Datei, in der der Parameterset enthalten ist.
Standard: CFS.JRSAVE.*xxxx*. Näheres hierzu siehe Seite **565**.

? Die FT-Maske wird mit dem Inhalt des Parametersets gefüllt und dem Benutzer angezeigt. Standardmäßig wird das FT-Kommando sofort ausgeführt.

Hinweise:

Die durch den Parameterset aktivierten File-Transfer Parameter können durch nachfolgend angegebene Parameterwerte wie *datei1*, *datei2* modifiziert werden.

Das Abspeichern der zuletzt benutzten FT-Parameter in einen Parameterset erfolgt in der Dateienliste von CFS mit dem Kommando SP FT%*name*. Näheres hierzu auf Seite **283**.

datei1 Name der zu übertragenden Datei/des Bibliothekselements (PLAM).
Ein Bibliothekselement ist anzugeben in der Form: *LIB(*bibl,element,typ*)

BS2000 Posix-Dateinamen sind in Hochkommas anzugeben. Die Namen werden in diesem Fall nicht in Großbuchstaben umgesetzt.

datei2 Name der Datei/der PLAM-Bibliothek im Zielsystem.
Ein Bibliothekselement ist anzugeben in der Form: *LIB(*bibl,element,typ*)
BS2000 Posix-Dateinamen sind in Hochkommas anzugeben. Die Namen werden in diesem Fall nicht in Großbuchstaben umgesetzt.
Standard: gleicher Name wie *datei1*.

passw Passwort der zu überschreibenden/abzuholenden Datei im Remote-System
[C]'...' | X'...'

host2 Symbolischer Name des fremden Host-Rechners.
host2 ist der Partnername laut FT-Beschreibung im aufrufenden Systems und nicht der Host-Name laut BCAM-Generierung.

user-id2 User-Id für die Transfer-Admission im fremden Host-Rechner.
Standard: gleiche Benutzerkennung wie im eigenen Host.

FTAC: Alternativ zu einer Remote User-Id kann auch eine FTAC Transfer-Admission für das fremde System angegeben werden. Hierzu ist im Feld "Remote User-ID" die Konstante "FTAC:" anzugeben, gefolgt von der eigentlichen Transfer-Admission. Bei der Eingabe der Transfer-Admission ist auf korrekte Groß-/Kleinschreibung entsprechend dem FTAC-Profil zu achten.

Hinweis:

Der Systemverwalter muß CFS bei der Installation bekannt machen, ob im eigenen Host das Produkt FTAC verfügbar ist oder nicht. Dies geschieht durch eine entsprechende Angabe im Modul CFSMAIN (Standardwert: FTAC ist nicht verfügbar). Siehe hierzu CFS-Handbuch für Systemverwalter, Abschnitt "Installation von CFS".

acct-nr2 Abrechnungsnummer.
Standard: gleiche Abrechnungsnummer wie im eigenen Host.

passw2 Logon-Passwort der Benutzerkennung im fremden System ([C]'...' | X'...').
Standard: kein Passwort.

rem-succ Remote Success-Procedure. Folgeverarbeitung im Remote-System, die bei erfolgreicher Übertragung ausgeführt wird.

*proc,** Die als *proc* angegebene DO-Prozedur erhält als Parameter den Namen der übertragenen Datei.

** Bei einer Übertragung mit einem FTAC-Profil wird für die Success-Behandlung nur der Dateiname übergeben.

/cmd .... Nach erfolgreicher Übertragung wird im Remote-System das BS2000-Kommando ausgeführt. Falls '*' im Kommando angegeben wurde, so wird an der entsprechenden Stelle der Name der Remote-Datei substituiert. Auf '*' dürfen keine weiteren Parameter für das BS2000-Kommando *cmd* folgen. Falls notwendig, ist hierzu die Variante *proc,** zu verwenden. Es können mehrere Kommandos, getrennt durch Semikolons, angegeben werden (/*cmd1* ...;/*cmd2* ...).

'/cmd ...' Nach erfolgreicher Übertragung wird im Remote-System das angegebene Kommando ausgeführt. Das Zeichen '*' wird in dieser Variante nicht durch einen Dateinamen ersetzt.

Standard: keine Remote Success-Procedure.

rem-fail Remote Failure-Procedure. Name einer Folgeverarbeitung im Remote-System, die bei nicht erfolgreicher Übertragung ausgeführt wird.

*proc,** Die als *proc* angegebene DO-Prozedur erhält als Parameter den Namen der zu übertragenden Datei.

* Bei einer Übertragung mit einem FTAC-Profil wird für die Failure-Behandlung nur der Dateiname übergeben.

/cmd .... Bei nicht erfolgreicher Übertragung wird im Remote-System das BS2000-Kommando ausgeführt. Falls '*' im Kommando angegeben wurde, so wird an der entsprechenden Stelle der Name der Remote-Datei substituiert. Auf '*' dürfen keine weiteren Parameter für das Kommando *cmd* folgen. Falls notwendig, ist hierzu *proc,** (siehe oben) zu verwenden. Es können mehrere Kommandos, getrennt durch Semikolons, angegeben werden (*/cmd1*...;*/cmd2* ...).

'/cmd ...' Bei nicht erfolgreicher Übertragung wird im Remote-System das angegebene Kommando ausgeführt. Das Zeichen '*' wird in dieser Variante nicht durch einen Dateinamen ersetzt.

Standard: keine Remote Failure-Procedure.

rem-par Remote Parameter (Typ des fremden Systems).

mögliche Angaben: <u>*BS2000</u> | *MSP | *ANY | *any

*BS2000 Zielrechner hat das Betriebssystem BS2000.
*MSP Zielrechner hat das Betriebssystem MVS.
*ANY Zielrechner hat ein anderes Betriebssystem (SINIX/VMS/..). Dateinamen werden dem Fremdsystem in <u>Großbuchstaben</u> übergeben.
*any wie *ANY, jedoch werden die BS2000-Dateinamen dem Fremdsystem in <u>Kleinbuchstaben</u> übergeben.
Standard: *BS2000.

direct Transfer-Direction <u>TO</u> | FROM (Richtung des File-Transfer).
Standard: TO.

lst Soll für jede übertragene Datei ein Protokoll ausgegeben werden ? YES | <u>NO</u>
Die Angabe YES bewirkt außerdem, daß bei Beendigung des File-Transfer Auftrags das Job-Report System von CFS aktiviert wird, um die von FT erzeugte Ergebnisliste am Bildschirm anzuzeigen.
Standard: NO.

Hinweis: Die von openFT ausgegebene Meldung "FILE TRANSFERRED" ist bei Kontrolle durch das Job-Report System überflüssig und kann durch das Kommando /BCNTRL MES=NO abgeschaltet werden.

time *hh:mm* Beginn der Übertragung nicht vor der angegebenen Uhrzeit.
Wurde eine Startzeit <u>vor</u> der aktuellen Uhrzeit angegeben, so wird der Auftrag am folgenden Tag zur angegebenen Uhrzeit angestartet.
Standard: Beginn der Übertragung zum frühestmöglichen Zeitpunkt.

" Modify Extended Parameters for FT " (letztes Feld der FT-Maske):

Durch die Eingabe Y in diesem Feld wird in die File-Transfer Extended Parameter Maske (FTEP) verzweigt. In dieser können zusätzliche, jedoch weniger häufig benötigte Parameter für alle folgenden FT-Aufrufe vorbelegt werden. dazu gehören u.a. die

Parameter "Local Transfer Admission User-Id, Acct-Nr, Password", sowie "Local Success Processing" und "Local Failure Processing". Die zusätzlichen FT-Parameter können auch vor dem ersten FT-Aufruf durch das Kommando FTEP [*param*] (siehe Seite **244**) vorbelegt werden.
Standard: N (kein Verzweigen in die FTEP-Maske)

Hinweise:

Ab openFT Version 3 kann die Beendigung von FT-Aufträgen mit dem Job-Report System von CFS überwacht werden (FT-Parameter Protocol-Listing: YES). Weitere Informationen zu diesem Thema auf Seite **412**.

Bei der Übertragung von Dateien mit dem Kommando FT werden im lokalen System Read-Passworte für die zu übertragenden Dateien dynamisch nachgefordert.

Zur Selektion aller mit openFT übertragenen Dateien wird auch auf die NSTAT-Option im Feld FILENAME-SELECT der Selektionsmaske hingewiesen.

Werden die Parameter des FT-Kommandos bis einschließlich *acct-nr2* im FT-Kommando explizit angegeben, so wird die FT-Parameter Maske nicht ausgegeben und für evtl. noch folgende Parameter werden die angegebenen Standardwerte angenommen. In allen anderen Fällen wird die FT-Maske ausgegeben, in der alle Parameter eingetragen werden können. Um eine Maskenanforderung zu vermeiden, sollte das Kommando FT in Prozeduren mindestens mit den folgenden Parametern angegeben werden: FT *datei1*, [*datei2*], [*passw*], *host*, *user-id2*, [*acct-nr2*]

Nach Absenden der FT-Maske und nach Ausführung des FT-Kommandos wird die FT-Maske wieder angezeigt zur Übertragung einer weiteren Datei. Dies geschieht solange, bis der Benutzer in der FT-Maske bei *datei1* Blanks einträgt oder die FT-Maske mit der Taste **K1** sendet.

Durch Betätigen der K3-Taste kann, ähnlich wie im EDT, ein Bildschirm-Refresh der FT-Maske erreicht werden.

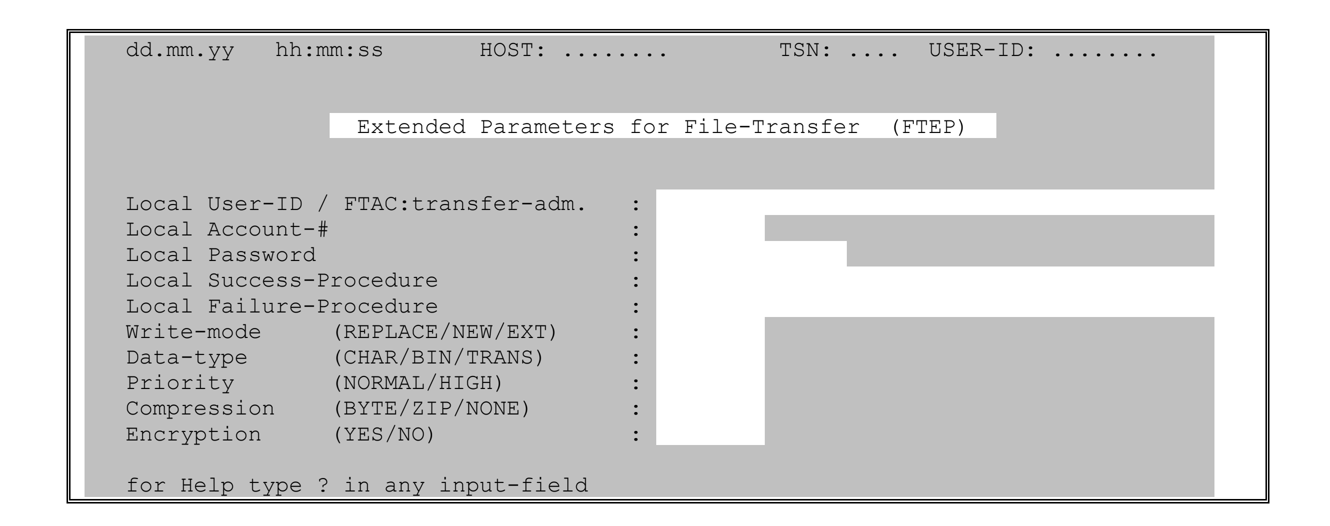
Für weitere Hinweise zum File-Transfer - insbesondere File-Transfer zwischen BS2000 und SINIX-Systemen - wird auf das Kapitel 12 "File-Transfer mit openFT/RDAC" verwiesen.

Zusätzliche File-Transfer Parameter anzeigen/ändern

FTEP [*user-id*] [*,acct-nr*] [*,passw*] [*,loc-succ*] [*,loc-fail*] [*,wr-mode*] [*,data-typ*] [*,prio*] [*,comp*] [*,encrypt*]

Die FTEP-Parameter dienen als Hintergrundinformation für alle folgenden FT-Aufrufe. Gesetzte FTEP-Parameter können durch ein erneutes Verzweigen in die FTEP-Maske und Eintragen von Blanks zurückgesetzt, bzw. durch andere Werte ersetzt werden.

Alle Parameter des FTEP-Kommandos sind wahlweise. Falls keine Parameter angegeben sind, so werden diese in der FTEP-Maske angefordert:



```
dd.mm.yy   hh:mm:ss       HOST: ........        TSN: ....  USER-ID: ........

                   Extended Parameters for File-Transfer  (FTEP)

Local User-ID / FTAC:transfer-adm.   :
Local Account-#                      :
Local Password                       :
Local Success-Procedure              :
Local Failure-Procedure              :
Write-mode      (REPLACE/NEW/EXT)    :
Data-type       (CHAR/BIN/TRANS)     :
Priority        (NORMAL/HIGH)        :
Compression     (BYTE/ZIP/NONE)      :
Encryption      (YES/NO)             :

for Help type ? in any input-field
```

Parameter des FTEP-Kommandos:

user-id User-Id für die Transfer-Admission und - sofern eine *loc-succ*, bzw. *loc-fail* Prozedur angegeben - für die Processing-Admission im lokalen System (eigener Host).
Standard: eigene Benutzerkennung.

Alternativ zu der lokalen User-Id kann auch eine **FTAC** Transfer-Admission für das lokale System angegeben werden. Hierzu ist im Feld "Local User-ID" die Konstante 'FTAC:', gefolgt von der FTAC Transfer-Admission anzugeben.

acct-nr Abrechnungsnummer der unter User-Id angegebenen Benutzerkennung.
Standard: Abrechnungsnummer der eigenen Kennung.

passw Logon-Passwort der Benutzerkennung für die lokale Transfer-Admission.
Standard: kein Passwort.

loc-succ Local Success-Procedure. Name einer Folgeverarbeitung, die bei erfolgreicher Übertragung im lokalen System (eigener Host) ausgeführt wird.

*proc,** Die als *proc* angegebene DO-Prozedur erhält als Parameter den Namen der zu übertragenden Datei.

* Bei einer Übertragung mit einem FTAC-Profil wird für die Success-Behandlung nur der Dateiname übergeben.

/cmd .... Nach erfolgreicher Übertragung wird im lokalen System das BS2000-Kommando ausgeführt. Falls '*' im Kommando angegeben wurde, so wird an der entsprechenden Stelle der Name der lokalen Datei substituiert. Auf '*' dürfen keine weiteren Parameter für das Kommando *cmd* folgen. Falls notwendig, ist hierzu die Variante *proc,** zu verwenden. Es können auch mehrere Kommandos, getrennt durch Semikolons angegeben werden (/*cmd1* ...; /*cmd2* ...).

'/cmd ...' Nach erfolgreicher Übertragung wird im lokalen System das angegebene Kommando ausgeführt. Das Zeichen '*' wird in dieser Variante nicht durch einen Dateinamen ersetzt.
Standard: keine Local Success-Procedure.

loc-fail Local Failure-Procedure. Name einer Folgeverarbeitung, die bei nicht erfolgreicher Übertragung im lokalen System (eigener Host) ausgeführt wird.

*proc,** Die als *proc* angegebene DO-Prozedur erhält als Parameter den Namen der zu übertragenden Datei.

* Bei einer Übertragung mit einem FTAC-Profil wird für die Failure-Behandlung nur der Dateiname übergeben.

/cmd .... Bei nicht erfolgreicher Übertragung wird im lokalen System das BS2000-Kommando ausgeführt. Falls '*' im Kommando angegeben wurde, so wird an der entsprechenden Stelle der Name der lokalen Datei substituiert. Auf '*' dürfen keine weiteren Parameter für das Kommando *cmd* folgen. Falls notwendig, ist hierzu *proc,** (siehe oben) zu verwenden.

'/cmd ...' Bei nicht erfolgreicher Übertragung wird im lokalen System das angegebene Kommando ausgeführt. Das Zeichen '*' wird in dieser Variante nicht durch einen Dateinamen ersetzt.
Standard: keine Local Failure-Procedure.

wr-mode Write-Modus. Hiermit wird festgelegt, wie bei einer bereits bestehenden Empfangsdatei zu verfahren ist.

REPLACE Existiert im Empfangssystem eine gleichnamige Datei, so wird diese überschrieben. (Standard)

NEW Existiert im Empfangssystem eine gleichnamige Datei, so wird diese nicht überschrieben und die Datei wird nicht übertragen.

EXT Extend. Eine bereits bestehende Empfangsdatei wird im Extend-Modus eröffnet (erweitert).

Standard: REPLACE

data-typ Data-Type. Hiermit wird angegeben, wie die Daten in der Sendedatei codiert sind.

CHAR Die Sendedatei ist eine Textdatei. Das Empfangssystem speichert die Datei in seinem Zeichencode als Text ab. Falls notwendig, wird die Datei umcodiert. (Standard)

BIN Die Sendedatei ist eine Binärdatei. Das Empfangssystem speichert die Datei so ab, wie sie vom Sendesystem geliefert wird. Eine Umcodierung findet nicht statt.

TRANS Transparente binäre Datenübertragung (DATA-TYPE= *BIN(TRANS= *YES)). Beliebige BS2000-Dateien, insbesondere auch ISAM- und PAM-Dateien können über Nicht-BS2000 Zwischenstationen (z.B. UNIX oder PC) so übertragen werden, daß sie in dem entfernten BS2000 Zielsystem wieder die gleichen Dateimerkmale wie auf dem Sendesystem besitzen.
Die Option TRANS steht ab openFT V6.0 zur Verfügung.

Standard: Der Typ der Sendedatei ist nicht bekannt und wird im fernen System bestimmt.

prio Gibt die Priorität an, mit der die Dateiübertragung im Vergleich zu anderen Dateiübertragungen mit demselben fernen System gestartet wird.

NORMAL Die Dateiübertragung hat normale Priorität (Standard).

HIGH Die Dateiübertragung hat hohe Priorität.

comp Komprimierung während der Übertragung.

BYTE Die Datei wird mit Bytekomprimierung übertragen (Standard).

ZIP Die Datei wird mit ZIP Komprimierungsalgorithmus übertragen (ab OpenFT V10.0).

NONE Die Datei wird ohne Komprimierung übertragen.
Dieser Modus kann z.B. bei der Übertragung in SINIX-Systeme notwendig sein.

encrypt Verschlüsselte Dateiübertragung (ab openFT V6.0).

YES Die Datei wird verschlüsselt übertragen. Diese Funktion setzt das Produkt openFT-CR voraus.

NO Die Datei wird ohne Verschlüsselung übertragen (Standard).

Guard-Schutz für Dateien / Bibliothekselemente festlegen

GUARD [%*name* | %? [,JRS=*jrfile*]] [*datei*] [,R=*guard*|*NONE|*SAME] [,W=*guard*|*NONE| *SAME] [,E=*guard*|*NONE|*SAME]

Die Datei bzw. das PLAM-Bibliothekselement mit dem angegebenen Namen wird mit einem Guard-Schutz versehen. Falls keine Parameter bzw. nur der Dateiname angegeben ist, wird eine Maske mit dem folgenden Format ausgegeben:

```
dd.mm.yy   hh:mm:ss       HOST: ........        TSN: ....  USER-ID: ........


                              modify GUARDS-protection

Filename                     :


Read-Guard  (guard-name/*NONE/*SAME)  :
Write-Guard (guard-name/*NONE/*SAME)  :
Exec-Guard  (guard-name/*NONE/*SAME)  :


for Help type ? in any input-field
```

%*name* Die Parameterangaben für das Kommando GUARD werden dem angegebenen Parameterset entnommen. Die Speicherung des Parametersets erfolgt mit dem CFS-Kommando SPGUARD %*name*.

%? Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

JRS=*jrfile* Name der JRSAVE-Datei, in der der Parameterset gespeichert ist.
Standard: CFS.JRSAVE.*xxxx*. Näheres hierzu siehe Seite 565.

datei	Name der zu schützenden Datei bzw. des zu schützenden Bibliothekselements. Der Name eines Bibliothekselements wird in der Form *bibl*([*t*/]*name*[/*vers*]) angegeben. Anstelle eines Namens kann auch die Kurzbezeichnung * angegeben werden. Das gewünschte Datenobjekt muß hierbei mit dem Action-Code * markiert werden. Falls kein Name angegeben bzw. in der Maske eingetragen wurde, wird keine Aktion ausgeführt. Die angegebenen Guard-Profile werden jedoch für eine spätere Verwendung mit dem Action-Code SG (Set Guard) gemerkt.
R=*guard*	Der lesende Zugriff auf das Datenobjekt wird mit einem Guard des angegebenen Namens geschützt. Ein Guard kann wie eine Datei mit PVS-ID, User-ID und Name angegeben werden. Der Name ist auf acht Stellen begrenzt.
R=*NONE	Datei: Der lesende Zugriff wird mit einem leeren Guard geschützt und ist daher nicht möglich. PLAM-Element: Der lesende Zugriff wird weder durch Guard-, noch durch Basic-Access (BACL) Zugriffskontrollen überwacht.
R=*SAME	Die Rechte des lesenden Zugriffs auf das Datenobjekt werden nicht verändert.
W=*guard*	Der schreibende Zugriff auf das Datenobjekt wird mit einem Guard des angegebenen Namens geschützt. Ein Guard kann wie eine Datei mit PVS-ID, User-ID und Name angegeben werden. Der Name ist auf acht Stellen begrenzt.
W=*NONE	Datei: Der schreibende Zugriff wird mit einem leeren Guard geschützt und ist daher nicht möglich. PLAM-Element: Der schreibende Zugriff wird weder durch Guard-, noch durch Basic-Access (BACL) Zugriffskontrollen überwacht.
W=*SAME	Die Rechte des schreibenden Zugriffs auf das Datenobjekt werden nicht verändert.
E=*guard*	Der ausführende Zugriff auf das Datenobjekt wird mit einem Guard des angegebenen Namens geschützt. Ein Guard kann wie eine Datei mit PVS-ID, User-ID und Name angegeben werden. Der Name ist auf acht Stellen begrenzt.
E=*NONE	Datei: Der ausführende Zugriff wird mit einem leeren Guard geschützt und ist daher nicht möglich. PLAM-Element: Der ausführende Zugriff wird weder durch Guard-, noch durch Basic-Access (BACL) Zugriffskontrollen überwacht.
E=*SAME	Die Rechte des ausführenden Zugriffs auf das Datenobjekt werden nicht verändert.

Hinweise:

Für Dateien gilt folgende Vereinbarung: Falls bei allen drei Zugriffsarten (R/W/E) der Wert *NONE eingetragen ist, wird die Guard-Zugriffskontrolle insgesamt ausgeschaltet. Die Dateien sind danach nur noch mit SHARE=YES|NO,ACCESS=READ|WRITE geschützt. Der gleiche Effekt kann auch mit dem Action-Code SGN (Set Guards None) erreicht werden.

Bei dem Schutz von PLAM-Elementen mit dem Kommando GUARD wird die Existenz der angegebenen Guards nicht geprüft. Wird eine Zugriffsart, z.B. R (Read) mit einem nicht existierenden Guard geschützt, so ist der entsprechende Zugriff (z.B. Read) für alle Benutzer solange gesperrt, bis ein Guard mit dem angegebenen Namen eingerichtet wird.

Hardcopy-Modus einschalten

HC [*n*] [*datei*] [, L] [, T] [, B] [, E] [, O] [, A] [, F] [, G] [, I] [, S] [, R] [, RS] [, SMO]

Hardcopy-Modus für die CFS-Grundebene/für Connection *n* einschalten.
Bei eingeschaltetem Hardcopy-Modus werden die anfallenden Bildschirm Ein-/ Ausgaben (insbesondere Masken) bildschirmgerecht in einer Datei protokolliert.

Alle Operanden des HC-Kommandos sind wahlweise und können weggelassen werden. Es treten dann die entsprechenden Standardannahmen in Kraft.

Der Hardcopy-Modus kann beliebig oft ein- und wieder ausgeschaltet werden (Kommando NHC siehe unten).

Ohne den Hardcopy-Modus über das HC-Kommando einzuschalten, können einzelne CFS-Masken auch protokolliert werden, indem sie in der CFS-Grundebene mit der F3-Taste und in einer Connection mit einer vorher definierten Hardcopy-Taste (siehe Kommando CH, Seite 470) abgesendet werden. Es ist jedoch zu beachten, daß jede mit **F3** protokollierte Maske im ausgedruckten Listing auf einer neuen Seite erscheint. Im normalen Hardcopy-Modus (HC-Kommando) werden nach Möglichkeit mehrere Masken auf einer Druckseite untergebracht. Über ein File-Kommando mit LINK=HARDCOPY kann eine Hardcopy-Datei mit einem vorgegebenen Namen zugewiesen werden.

Das Absenden einer CFS-Maske mit **F3** wird, abgesehen von der Protokollierung wie ENTER behandelt. Beim Hardcopy einer Maske in einer Connection mit der durch das Kommando CH definierten Datenübertragungstaste gehen die Eingabedaten verloren und müssen noch einmal eingegeben und mit der ENTER-Taste abgesendet werden.

Das Kommando HC wirkt nur für den Basisprozeß von CFS und nicht für evtl. durch OC eröffnete Connections. Für diese muß ein HC*n*-Kommando angegeben werden. Der Hardcopy-Modus für eine Connection kann bereits vor dem Eröffnen einer Connection durch das OC-Kommando eingeschaltet werden. Innerhalb der Connection kann der Hardcopy-Modus durch das HC-Kommando angefordert werden.

Für eine ausführliche Beschreibung aller Operanden des HC-Kommandos wird auf Seite 439 und folgende (Hardcopy) verwiesen.

NHC | NHC *n* — Hardcopy-Modus für CFS-Grundebene / für Connection *n* ausschalten.

Eintrag in Dateienliste hinzufügen

I *datei* [, *act*] — Include. Der Dateienliste wird die angegebene Datei/Jobvariable hinzugefügt. Es kann ein sowohl vollqualifizierter, als auch ein teilqualifizierter Name *teilqual.*, oder ein Suchmuster mit Wildcard-Zeichen nach FSTAT-Syntax *abc***xyz* angegeben werden. Falls es sich bei *datei* um eine shareable Datei unter einer anderen Kennung handelt, so ist der Name der Datei in der Form $*user-id. datei* anzugeben.

act — der Action-Code wird bei der hinzugefügten einzelnen Datei eingetragen.
Der Action-Code D (Display) wirkt in der Weise, daß die Datei der Liste hinzugefügt und sofort zum Display dieser Datei übergegangen wird.

Zum Thema "Hinzufügen von Dateien" siehe auch Kommando AL (Append List).

Dateiinformationssystem: Standard-Format für die Erfassung von INF-Texten

I FORM=*f*

Mit diesem Kommando kann ein benutzerindividuelles Format *f* im zentralen Formatkatalog $TSOS.CFS.INF.FORMAT definiert werden, das standardmäßig bei der Neuerfassung von Dokumentationen verwendet wird (Action-Codes IM/IA). Für weitere Informationen siehe Seite 421 und folgende (Datei-Informationssystem).

Termindatei (Item-File) zuweisen

IFILE *datei*

Item-File. Mit diesem Kommando legt der Benutzer eine Termindatei fest. Die Termindatei wird benötigt für die Kommandos CAL (Calendar), SI (Show Item), SIS (Show Item Short), SIL (Show Item List), MI/II (Modify/Insert Item).

Eine ausführliche Beschreibung des Kommandos finden Sie auf Seite 427 und folgende (Terminverwaltungssystem).

Insert Item: Neuen Termin eingeben

II [*datum*] [*/time*]

Insert Item. Es wird eine Leermaske zur Eingabe der Kurz- und Langinformationen für den Termin angezeigt.

Eine ausführliche Beschreibung des Kommandos finden Sie auf Seite 427 und folgende (Terminverwaltungssystem).

Action-Code in alle Action-Felder eintragen

INSRT *act*

Bei allen Elementen der Dateienliste wird im Action-Feld der angegebene Code *act* eingetragen. Das Sichtfenster wird wieder zum Anfang der Dateienliste positioniert.

Beispiel:

```
/PROC C
/SYSFILE SYSDTA=(SYSCMD)
/SETSW ON=(1,4)
/EXEC $CFS.CFS
*002 FILE=SYS.CONSLOG.
ONXADD LIB.CONSLOG
INSRT XE;AT
/SETSW OFF=(1,4)
/ENDP
```

Hinweis:

Eine weitere Möglichkeit, bei Elementen der Dateienliste automatisch einen vorgegebenen Action-Code einzutragen, bietet das Suche-Kommando in der Form S,'...' =INSRT .. . Für eine ausführliche Beschreibung siehe Seite 277.

Kommandos zu Steuerung des Job-Report Systems

JR.. JR | JREP | JRF | JRFT | JRIC | JRID | JRP | JRSH

Diese Kommandos finden Sie ausführlich beschrieben auf Seite **410** und folgende.

Letztes Kommando nicht löschen

KC | NKC Keep Command /do Not Keep Command.

KC Das zuletzt eingegebene Kommando wird im Kommandofeld nicht gelöscht.
NKC Das zuletzt eingegebene Kommando wird bei korrekter Ausführung gelöscht.
Standard: NKC

Zum Thema "Letztes Kommando wiederholen", siehe auch Seite **211** (Gedächtnis der eingegebenen Kommandos).

Last Page als Merkmal der Dateigröße

LASTP In der ersten Spalte der Dateienliste wird als Merkmal für die Größe der Datei der Wert Last Page angezeigt, d.h. die Anzahl der tatsächlich beschriebenen PAM-Blöcke.
Standard: LASTP.

Eingaben [nicht] im Kommandogedächtnis ablegen

[N]LIM [SEL|CMD|CON]

Log Input in Memory (LIM) / do Not Log Input in Memory (NLIM).

LIM Alle Benutzereingaben (Eingaben in der Selektionsmaske, Kommandos, Line-Mode Eingaben in Connections) werden im internen Kommandogedächtnis protokolliert (Standard).

SEL Das Gedächtnis für Eingaben in der Selektionsmaske wird aktiviert.

CMD Das Gedächtnis für Eingaben in der Dateienliste (Feld "COMMAND :") wird aktiviert.

CON Das Gedächtnis für Eingaben in Connections wird aktiviert.

NLIM Alle nach diesem Kommando getätigten Eingaben werden nicht im internen Kommandogedächtnis protokolliert und stehen somit bei einer späteren DUE2-Anforderung nicht zur Verfügung. Das Kommando NLIM ist besonders in CFS-Prozeduren sinnvoll, da hiermit die von der Prozedur erzeugten Eingaben nicht in das Kommandogedächtnis aufgenommen werden.

SEL Kein Gedächtnis für Eingaben in der Selektionsmaske.

CMD Kein Gedächtnis für Eingaben in der Dateienliste (Feld "COMMAND :").

CON Kein Gedächtnis für Eingaben in Connections.

Hinweis:

Im Initialisierungsmodul CFSMAIN kann vorgegeben werden, welche Eingaben in das Kommandogedächtnis aufgenommen werden sollen. Für weitere Informationen zum Thema Kommandogedächtnis siehe Seite **211**.

Kommandogedächtnis aus Datei laden

LM [*datei* | [*bibl*] (*t*/ *element*)] [, C]

Load Memory. Das CFS-interne Gedächtnis (Eingaben in der Selektionsmaske, Kommandoeingaben, Line-Mode Eingaben in Connections) wird mit dem Inhalt der angegebenen Datei bzw. mit dem Inhalt des Bibliothekselements geladen.

datei Name der Datei, aus der das Kommandogedächtnis geladen wird.

[*bibl*] ([*t*/] *element*)

Name der Bibliothek und des Elements, aus der das Kommandogedächtnis geladen wird. Standard für die Bibliothek: CFS-Userlib. Standard für den Elementtyp: S (Source). Für eine ausführliche Beschreibung der Userlib siehe Seite **541** und folgende.

Wird im Kommando LM weder ein Datei-, noch ein Elementname angegeben, so wird das Kommandogedächtnis aus einer Datei mit dem Namen CFS.MEM.STD.*jobname* geladen, bzw. CFS.MEM.STD.*acct*, falls beim LOGON kein Jobname angegeben wurde.

C Clear. Normalerweise werden die bereits bestehenden internen Tabellen durch die in der Datei bzw. in dem Bibliothekselement enthaltenen Eingaben ergänzt. Die C-Option bewirkt, daß die internen Tabellen für das Kommandogedächtnis vor Ausführung des LM-Kommandos zurückgesetzt werden.

Hinweise:

Für das Kommando LM *datei* gilt folgende Vereinbarung:
Falls *datei* nicht existiert (FSTAT-Returncode 0533), versucht CFS, den Inhalt des Elements *datei* (bei PLAM-Bibliotheken S/*datei*) aus der CFS-Userlib zu laden.

Für weitere Informationen zum Thema Kommandogedächtnis siehe Seite **211**.

Aufruf von LMS als Unterprogramm

LMS In LMS verzweigen (ISP-Oberfläche, Prompt = '$').

LMSSDF In LMS verzweigen (SDF-Oberfläche, Prompt = '//').

Rückkehr durch LMS-Anweisung END.

Nach dem Verlassen und erneuten Aufruf des LMS, so ist folgendes zu beachten:

- Die beim ersten LMS-Aufruf angesprochene Benutzeroberfläche (ISP/SDF) bleibt für alle folgenden Aufrufe gültig.
- Zuvor gegebene LIB-Anweisungen müssen nicht wiederholt werden. Sie gelten vom vorhergehenden LMS-Aufruf weiter.

Sperren des Bildschirms gegen unbefugte Benutzung

LOCK [*passw*] Der Bildschirm, an dem CFS läuft, wird gesperrt.
Aufgrund des LOCK-Kommandos wird von CFS eine Maske am Bildschirm ausgegeben (LOCK-Maske), die ein achtstelliges dunkelgesteuertes Eingabefeld enthält. Um den LOCK-Status zu beenden (Unlock), muß der Benutzer im Eingabefeld der LOCK-Maske das vereinbarte Passwort angeben. Im LOCK-Zustand ist es nicht möglich, über irgendwelche Funktionen von CFS zu verfügen; es ist auch nicht möglich, CFS im Basisprozeß durch Betätigung der **K2**-Taste zu unterbrechen und BS2000-Kommandos einzugeben.

passw Paßwort zum Entsperren des Bildschirms.
Beliebige ein- bis achtstellige alphanumerische Zeichenfolge.

Hinweise:

Zum Vereinbaren eines LOCK-Passworts siehe auch Kommando SPL (Set Password for Lock) Seite 482.

Wurde im Kommando LOCK kein Passwort angegeben, so gilt das in einem früheren LOCK-Kommando zuletzt vereinbarte Passwort bzw. das im Kommando SPL (Set Password for Lock) vereinbarte Kennwort.

Falls das Programm CFS in einer Connection noch einmal geladen wurde, so ist bei dem Kommando LOCK folgendes zu beachten: Die Eingabesperre bezieht sich nur auf das in der Connection geladene CFS.

Das LOCK-Kommando trägt dem Bedürfnis nach Datenschutz jedes einzelnen Benutzers Rechnung: Der Benutzer kann sich von seinem Bildschirm entfernen, ohne daß eine andere Person in der Lage wäre, mit dem noch geladenen CFS zu arbeiten oder durch das Kommandogedächtnis sich über die eingegebenen Kommandos oder die eröffneten Connections zu informieren.

In der LOCK-Maske ist der Bereich ab der dritten Bildschirmzeile modifizierbar. Der Benutzer kann in diesem Bereich eine Nachricht für seine Mitarbeiter hinterlassen.
Beispiel: "Bin bis 16:00 auf Besprechung (Raum C1/223)".
Durch eine Datenübertragung seitens eines fremden Benutzers wird diese Nachricht nicht gelöscht.

Kleinbuchstaben nicht in Großbuchstaben umwandeln

LOW CFS-Editor: Beim Modifizieren wird keine Umwandlung der eingegebenen Kleinbuchstaben in Großbuchstaben vorgenommen.

Eingabe von Dateidokumentationen (Action-Code IM): Der Text wird in der eingegebenen Groß-/Kleinschreibung übernommen.

Suche-Kommando, User Option FIND, Variable Action ONXFIND: Im Suchargument erfolgt keine Umwandlung von Kleinbuchstaben in Großbuchstaben. Das gleiche kann auch durch Eingabe des Sucharguments in der Form L'....' erreicht werden.

Action-Code EDT: Beim Einlesen einer neuen Datei/eines neuen Bibliothekselements wird der LOWER-Modus des EDT unverändert gelassen. Ein bei der Bearbeitung der letzten Datei eingegebenes LOWER ON-Kommando bleibt damit wirksam.

Job-Report Do-Parameter Maske: In Hochkommas eingegebene Parameter werden bezüglich der Groß- Kleinschreibung unverändert an die Prozedur übergeben und nicht, wie dies standardmäßig der Fall ist (CAP), in Großbuchstaben umgewandelt.

Speicherbereich in CFS modifizieren

M L'*addr*', C'*string*' | X'*string*'

Mit diesem Kommando können im geladenen Programm CFS bestimmte Speicherbereiche mit einem vom Standard abweichenden Wert überschrieben werden. Das Modify-Kommando ist ein Ersatz für das AID-Kommando %M C'...' INTO V'...'. Das Modify-Kommando sollte nur zum Ändern von CFS-internen Parametern verwendet werden, für die kein eigenes Kommando existiert.

L'*addr*' Adresse der zu modifizierenden Stelle im geladenen Programm CFS. Die Adressen bestimmter Parameter in CFS können aus dem Quellprogramm CFSMAIN ermittelt werden. Dem Assembler-Befehl MVI 33(R1),C'Y' entspricht in der CFS-Syntax z.B. das Kommando M L'2021',C'Y' (L'2021' = L'2000' + 33). Die in CFSMAIN angesprochenen Adressen CFS-interner Parameter werden sich auch in zukünftigen CFS-Versionen nicht ändern.

C'*string*' | X'*string*' An die angegebene Adresse zu übertragender Inhalt.

Beispiele:

```
M L'2031',X'00'
```

Mit dieser Anweisung wird der Inhalt des Bytes an der Adresse L'2031' auf den Wert X'00' gesetzt. Damit wird erreicht, daß beim Hardcopy von Masken in eine druckaufbereitete Datei Kleinbuchstaben von CFS nicht in Großbuchstaben umgesetzt werden.

```
M L'22E5',C'CFS.JRSAVE.name'
```

Mit dieser Anweisung wird der Inhalt des Speicherbereichs ab Adresse L'22E5' mit der Character-Zeichenfolge 'CFS.JRSAVE.name⊔ ⊔ ⊔' gefüllt. Damit wird ein von der Standard JRSAVE-Datei abweichender Name vereinbart, z.B. für zwei Benutzer unter der gleichen Kennung.

Rückkehr aus CFS in das aufrufende Hauptprogramm

MAIN In das Hauptprogramm zurückkehren.
Dieses Kommando ist nur sinnvoll, falls CFS als Unterprogramm von einem anderen Programm aus aufgerufen wurde.

Hinweise:

CFS ist unterprogrammfähig in dem Sinne, daß es als Großmodul von beliebigen Programmen aus aufgerufen werden kann (nachladbar oder fest eingebunden). Eine ausführliche Beschreibung der Schnittstelle finden Sie auf Seite 461.

CFS kann aufgerufen werden im Dialogmodus. Es meldet sich dann ebenso wie beim Aufruf per /EXEC-Kommando mit den gewohnten Masken.

CFS kann über die Unterprogrammschnittstelle auch im Prozedurmodus aufgerufen werden. In diesem Fall übergibt das aufrufende Programm CFS Anweisungsfolgen, die im Prozedurmodus von CFS ausgeführt werden. Die Anweisungsfolgen können über SYSDTA oder auch als Format-V Sätze im Speicher übergeben werden.

Mehr zu diesem Thema auf Seite 485 und folgende (Prozedursprache).

Mitteilungen, Änderungswünsche in CFS-Kommentardatei schreiben

MES

Mitteilung in CFS-Kommentardatei schreiben. Es wird eine Maske ausgegeben, in der Fehlermeldungen zu CFS, Änderungswünsche und sonstige Mitteilungen an die Fa. OPG eingetragen werden können. Diese Daten werden in die Datei $TSOS. CFSCOM geschrieben. Der Name der Datei kann bei der Installation vom Systemverwalter geändert werden.

Modify Item: Text zu Termin ändern

MI [*datum*]

Modify Item. Es wird der bisher eingegebene Text zu dem angegebenen Termin in einer Maske angezeigt und zur Änderung freigegeben.

Eine ausführliche Beschreibung des Kommandos finden Sie auf Seite 427 und folgende (Terminverwaltungssystem).

User Option in Dateienliste ändern

NL [*usopt1*] [;[*usopt2*]] [;[*usopt3*]] [;[*usopt4*]]

New List. Für die aktuell ausgewählten Datenobjekte der Dateienliste werden die Werte der angegebenen User Options angezeigt. Es wird dabei keine neue Selektion vorgenommen.

usopt1

Bezeichnung einer für den aktuellen Typ von Datenobjekten zugelassenen User Option. Mit der User Option angegebene Selektionsbedingungen werden ignoriert.

usopt2;*usopt3*;*usopt4*

Der Wert für die als erstes angegebene User Option (*usopt1*) wird in einer eigenen User Option Spalte in der Dateienliste eingetragen. Die Werte für eine evtl. angegebene zweite, dritte und vierte User Option (*usopt2*;*usopt3*;*usopt4*) überschreiben in der Dateienliste die Spalten PASS, AGE und LASTP (SPACE). Zwei aufeinanderfolgende Semikolons ";;" bewirken, daß die entsprechende Spalte der Dateienliste nicht mit einer User Option überschrieben wird. Siehe hierzu das zweite und dritte Beispiel.

Falls keine User Option angegeben ist, wird das Standard-Layout der Dateienliste ohne User Option angezeigt.

Beispiele:

```
NL ALLOC
```
Bei allen Dateien der Dateienliste wird der aktuelle Wert der User Option ALLOC eingetragen.

```
NL ALCNT;EXTNT;;ALLOC
```
Hiermit wird die User Option Spalte mit der Anzahl der erfolgten Sekundärzuweisungen (ALCNT) und die Spalte PASS mit der Anzahl der Extents (EXTNT) gefüllt. Die Spalte AGE wird nicht überschrieben (;;). Die Spalte LASTP wird mit dem der Datei zugewiesenen Platz (ALLOC) gefüllt.

```
NL ;;;VSN
```
Diese Angabe bewirkt, daß in der ersten Spalte der Dateienliste (LASTP) die VSN der jeweiligen Datei angezeigt wird. Die Spalten AGE und PASS bleiben unverändert.

Neue Auswahl von Datenobjekten

NP New Parameters. Es wird dem Benutzer die Selektionsmaske angeboten für eine neue Auswahl von Datenobjekten. Die bisherige Dateienliste und alle evtl. noch zur Ausführung anstehenden Actions werden hiermit ungültig.
Das Drücken der K1-Taste hat die gleiche Wirkung wie das Kommando NP. Siehe hierzu auch Seite 208.

An dieser Stelle wird auch auf das Kommando AL hingewiesen. Mit AL wird eine bestehende Liste durch die in einer neuen Selektion gefundenen Dateien verlängert.

NP *param* Das Feld FILENAME-SELECT und andere Felder der Selektionsmaske werden mit dem in *param* angegebenen Inhalt gefüllt. Gleich anschließend an das Kommando NP wird die entsprechend den Selektionsangaben neu aufgebaute Dateienliste angezeigt. Der Transaktionsschritt "Ausfüllen der Selektionsmaske" wird damit übersprungen. Die Selektionseingaben stehen zudem als ein Kommando im Kommandogedächtnis und können über die Taste Shift ENTER (DUE2) wieder am Bildschirm angezeigt werden.

param [*filename-sel*] [; *user-id*] [; *keyw1*=...] [; *keyw2*=...] [...]

Inhalt, mit dem die Felder der Selektionsmaske gefüllt werden sollen. Der erste Stellungsparameter *filename-sel* wird in das Maskenfeld FILENAME-SELECT, der zweite Stellungsparameter *user-id* in das Maskenfeld USER-ID eingetragen. Alle weiteren Parameter sind Schlüsselwortparameter und werden in die bezeichneten Maskenfelder eingetragen.

Im folgenden sind die kürzestmöglichen Zuordnungen von Schlüsselworten zu den entsprechenden Maskenfeldern aufgeführt:

LA --> LAST PAGE
AG --> AGE
FCB --> FCBTYPE
VOL --> VOLUME
SH --> SHARE / NONSHARE
ACC --> ACCESS
PAS --> PASS
U --> USER OPTION
SO --> SORT OPTION
DOC --> DOCUMENTATION
VAR --> VARIABLE ACTION

Das Separatorzeichen ';' kann mit dem Kommando SEP auch in ein anderes Zeichen umdefiniert werden. Nähere Informationen hierzu Siehe Seite 465 und folgende (Parameter ändern).

Beispiele:

`NP SRC`
Alle Dateien, die die Zeichenfolge 'SRC' in ihrem Namen enthalten.

`NP CFS.,Q`
Alle Dateien, die mit der Teilqualifizierung 'CFS.' beginnen (Qualified Selection).

`NP SRC;XYZ`
Alle Dateien mit der Zeichenfolge 'SRC' unter der Kennung $XYZ.

`NP SRC;XYZ;AG=0`
Alle heute neu angelegten oder veränderten Dateien (AGE=0) unter der Kennung $XYZ, die die Zeichenfolge 'SRC' in ihrem Namen enthalten.

`NP;US=RETPD>0;VAR=ONX/CAT RETPD=0`
Alle Dateien unter der eigenen Kennung, die ihr Freigabedatum in der Zukunft haben (USER OPTION :RETPD>0). Mit diesem NP-Kommando wird zugleich eine Variable Action zum Rücksetzen der Retention-Period definiert (VAR=ONX/CAT RETPD=0). Die Ausführung der Variablen Action erfolgt nach dem Markieren der gewünschten Dateien (vgl. ON-Kommando, Action-Code X).

NP %*name* | %? [,*jrsave*] [,?] [;*param*]

Das Feld FILENAME-SELECT und andere Felder der Selektionsmaske werden mit dem in dem Parameterset gespeicherten Inhalt gefüllt. Anschließend wird die entsprechend den Selektionsangaben neu aufgebaute Dateienliste angezeigt.

%*name* Name des Parametersets mit der Selektionsbedingung.

%? Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

jrsave Name der JRSAVE-Datei, in der der angegebene Parameterset enthalten ist.
Standard: CFS.JRSAVE.*xxxx*. Näheres siehe Seite **565**.

? Die Selektionsmaske wird mit dem Inhalt des Parametersets gefüllt und dem Benutzer angezeigt. Standardmäßig wird die Selektion sofort ausgeführt.

param Mit Ausnahme von *filename-sel* können für *param* die gleichen Angaben gemacht werden wie bei NP *param* (siehe oben). Die in *param* angegebenen Werte überschreiben die entsprechenden Werte des Parametersets.

Beispiel:

`NP %XYZ,CFS.JRSAVE.1;TSOS;U=LACC`
Die im Parameterset %XYZ in der Datei CFS.JRSAVE.1 gespeicherte Selektionsbedingung wird im Feld USER-ID durch den Wert TSOS und im Feld USER OPTION durch LACC modifiziert. Das Ergebnis der Selektion wird dem Benutzer als Dateienliste angezeigt.

Weitere Varianten des NP-Kommandos

NP; Alle Dateien aus der eigenen Kennung selektieren.
NP; hat die gleiche Wirkung wie das Absenden der leeren Selektionsmaske.

NP* Diese Variante des NP-Kommandos bewirkt, daß die Dateienliste nach den zuletzt eingegebenen Selektionskriterien neu erzeugt wird.

NP*;*param* Die Dateienliste wird neu erzeugt nach den zuletzt eingegebenen Selektionsbedingungen, die durch *param* modifiziert werden können. Das Eingabeformat für *param* ist weiter oben beschrieben (NP *param*).

Hinweise:

Falls bei der letzten Selektion eine User Option angegeben wurde, so wird diese User Option auch wieder für den neuen Selektionsvorgang verwendet. Eine Änderung der Standard User Option durch das Kommando UO (Seite **483**), kommt bei der Neuselektion durch NP* nicht zur Wirkung. Bei einer Neuselektion von Dateien durch das Kommando NP; wird die Änderung der Standard User Option berücksichtigt.

Falls in der Dateienliste Bibliotheken (FMS/LMS/PLAM) angezeigt werden, so bewirkt der Action-Code NPF/NPL/NP, eingetragen bei der gewünschten Bibliothek dasselbe wie die Kommandos NP FMS *bibliothek* / NP LMS *bibliothek*.

Variable Action definieren

ONX | ON& { *var-act* | %*name* [,*jrfile*] [,?] } [, ENTER]

Variable Action definieren.

ONX... Variable Action wird nur für die Elemente der Dateienliste ausgeführt, die mit dem Action-Code X markiert wurden. Das Markieren der gewünschten Objekte in der Dateienliste kann in einem Transaktionsschritt zusammen mit dem Senden des ONX-Kommandos erfolgen, wahlweise jedoch auch erst vor der nächsten Eingabe.

ON&... Variable Action wird auf alle selektierten Datenobjekte angewendet, ohne daß diese mit einem Action-Code markiert werden müssen.
Beispiel: `ON&MOVE vsn/device`.

var-act Die hier zu definierende Variable Action entspricht der Eingabemöglichkeit im Feld VARIABLE ACTION der Selektionsmaske; Beschreibung siehe Seite **107** und folgende.

%*name* [,*jrfile*] [,?]

Die Variable Action ist in dem Parameter Set %*name* gespeichert. *jrfile* muß angegeben werden, falls der Parameter Set %*name* nicht in der Standard JRSAVE-Datei gespeichert ist. Das Speichern einer Variablen Action in einen Parameter Set erfolgt mit dem Kommando SP VAR%*name* [,*jrfile*]. Näheres siehe Seite **283**.
Durch ONX %*name* [,*jrsave*] ,? wird der Inhalt der in dem Parameter Set gespeicherten Variablen Action angezeigt.
Durch ONX %? werden alle Parameter Sets angezeigt, in denen Variable Actions gespeichert sind.
Für Beispiele zu Variablen Actions mit Parameter Sets siehe Seite **108**.

ENTER Die Variable Action wird im Dialog definiert und die gewünschten Dateien angekreuzt, die Ausführung der Variablen Action erfolgt jedoch im Enter. Der von CFS hierzu erzeugte Enter-Auftrag wird durch das Job-Report-System überwacht. Ausführliche Informationen hierzu siehe Seite **111**.

Hinweise:

Für den Fall, daß die Kommandozeile (71 Byte) für die Variable Action zu kurz ist, wird folgende Vorgehensweise empfohlen: Absenden der im Kommandofeld eingegebenen Variablen Action ON..., soweit dies in diesem Feld möglich ist. Durch das anschließende Kommando VAR wird direkt in das zweizeilige Feld "VARIABLE ACTION" der Selektionsmaske verzweigt. Die Variable Action kann dort in der vollen Länge ergänzt werden.

Durch Setzen des Prozeßschalters 14 (/SETSW ON=14) vor der Eingabe des ON-Kommandos kann die Protokollierung der Ausführungsmeldungen am Bildschirm ausgeschaltet werden. Alle Datenobjekte, für die die Variable Action fehlerfrei ausgeführt wurde, werden in diesem Modus in der Dateienliste unsichtbar gemacht (Action-Code '-'). Nach Ausführung der Variablen Action für alle Dateien und nach Eingabe von U auf die nachfolgende Terminierungsabfrage erhält der Benutzer eine aktualisierte Dateienliste, in der nur noch diejenigen Datenobjekte aufgeführt sind, für die die Variable Action nicht ausgeführt werden konnte. Durch das Kommando YANK (siehe Seite **288**) können die unsichtbaren Einträge der Dateienliste jedoch wieder sichtbar gemacht werden.

Aufruf von PERCON als Unterprogramm

PERCON In PERCON verzweigen (ISP-Oberfläche).

PERCONSDF In PERCON verzweigen (SDF-Oberfläche, Prompt = '//').

Rückkehr durch PERCON-Anweisung END.

P-Tasten laden

PL [*modul*] [, *maplib*]

P-Load. Laden der programmierbaren Tasten.

modul Name eines mit dem Softwareprodukt PLUS oder TLOAD erzeugten Moduls zur Belegung der P-Tasten.

maplib Name der LMR-Bibliothek, in der der PLUS-/TLOAD-Modul gespeichert ist. Wird die Angabe der Bibliothek weggelassen, so nimmt CFS die Standard-bibliothek PLUS. MAPLIB (ohne $user-id. !) bzw. eine mit LINK=MAPLIB zugewiesene Bibliothek.

PL ON | OFF Das Umladen der P-Tasten beim Wechseln zwischen verschiedenen Connections wird zugelassen bzw. unterdrückt.

Hinweise:

Falls *modul* und *maplib* nicht angegeben wurden (PL), so wird der Modul PSTD in der CFSLIB zur Belegung der P-Tasten herangezogen (PL PSTD,CFSLIB).

Der Modul PSTD enthält die P-Tastenbelegungen des EDT (identisch mit der EDT-Anweisung P-K [P-Tasten laden]).

In der CFS-Dateienliste und im CFS-Display/Editor kann mit den programmierbaren Tasten P4 und P5 gemäß der EDT-Tastenbelegung vorwärts und zurück geblättert werden. P10 bewirkt wie im EDT das Weiterblättern zum nächsten Treffer nach einem Suche-Kommando.

Eine weitere Möglichkeit des Ladens von P-Tasten über das Kommandogedächtnis ist auf Seite **211** beschrieben.

Volume-Informationen aus Archive-Directory anzeigen

POOL [AV] [, DIR=*directory*] [, *string*] [, FILE=*datei*] [, SORT=C|E|R|D|O [, A|D]]

In einer Maske werden die folgenden Datenträger-Informationen zu einem Archive-Directory angezeigt:

VSN und Device-Typ des Datenträgers, Name der auf dem Datenträger gespeicherten Sicherungsversionen, Eigentümer des Datenträgers (Archive-Pool oder Operator), Expiration-Date des Datenträgers bzw. der Sicherungsversion, Bemerkung zu dem Datenträger (z.B. Magnetband ist unbrauchbar und soll bei zukünftigen Sicherungen nicht mehr verwendet werden).

AV Es werden nur die Archivnummern (VSN's) der im Pool verfügbaren Datenträger ausgegeben. Verfügbar sind diejenigen Datenträger, die kein Expiration-Date besitzen bzw. deren Expiration-Date abgelaufen ist.
Standard: Es werden Informationen zu allen, d.h. zu belegten und zu freien Volumes ausgegeben.

directory Dateiname des Archive-Directories.
Falls das POOL-Kommando in einer Archive-Dateienliste (siehe Seite **42** und **53**) abgesetzt wird, so kann der Parameter DIR=*directory* weggelassen werden. In diesem Fall ergänzt CFS den Namen des gerade angezeigten Directories.

string Es werden in der Pool-Liste nur die Zeilen ausgegeben, deren Inhalt die Suchbedingung erfüllen. Anstelle eines einfachen Sucharguments '*string*' ist auch eine Kombination aus mehreren Suchargumenten mit den Verknüpfungszeichen */,/+ möglich.
Für eine ausführliche Beschreibung der Suchbedingungen siehe Seite **309**.
Beispiele für Suchbedingungen: :1:H04246,:1:H043
'S.9412'
'OPERATOR'+'-07-'
UNUSABLE

datei Die POOL-Einträge werden in eine Datei mit dem angegebenen Namen geschrieben. Es erfolgt keine Ausgabe der Maske.

SORT=*x*[,A | D]
Die POOL-Einträge des POOL-Kommandos werden nach dem angegebenen Kriterium sortiert:

C Sortierung nach Spalte "Current Use"
D Sortierung nach Spalte "Device"
E Sortierung nach Spalte "Exp.Date"
O Sortierung nach Spalte "Owner"
R Sortierung nach Spalte "Remark"
V Sortierung nach Spalte "VSN"

A|D Die Sortierung erfolgt in auf- bzw. absteigender Reihenfolge.

Die vom Kommando POOL ausgegebene Maske besitzt das folgende Format:

```
dd.mm.yy     hh:mm:ss        DIRECTORY POOL-LIST          hostname      TSN: ....
COMMAND :
DIRECTORY = xxxxxxxxxxxxxxxxxxxxxxx
VSN     DEVICE     CURRENT USE       OWNER       EXP.DATE     REMARK
H04245  TAPE-C4    S.940622.040133   OPERATOR    1995-07-27
H04246  TAPE-C4    S.940622.040133   OPERATOR    1995-07-27   **  UNUSABLE  **
H04268  TAPE-C4    S.941206.040309   POOL
H04269  TAPE-C4    S.941206.040309   POOL
H04270  TAPE-C4    AVAILABLE         POOL
H04271  TAPE-C4    AVAILABLE         POOL                     **  UNUSABLE  **
H04272  TAPE-C4    S.941212.052718   POOL        1995-09-16
H04273  TAPE-C4    S.941212.052718   POOL        1995-09-16
H04274  TAPE-C4    S.941212.052718   POOL        1995-09-16
List continues    (Total=66,Available=23)                                Return: K1
```

Im Kommandofeld der POOL-Maske können folgende Kommandos angegeben werden:

ADD= *vsn* | (*vsn*, ...) [, DEVICE=<u>T1600</u> | *device*]

Es werden Magnetbänder oder Kassetten in den Datenträgerpool des aktuellen Directories aufgenommen. CFS erzeugt aus dem Kommando eine ARCHIVE-Anweisung POOL DIR=...,ADD=vsn [, DEVICE=...]

REM[OVE]= *vsn* | (*vsn*, ...)

Es werden Magnetbänder oder Kassetten aus den Datenträgerpool des aktuellen Directories entfernt. CFS erzeugt aus dem angegebenen Kommando eine ARCHIVE-Anweisung POOL DIR=...,REMOVE=...

PU[RGE] [SV= OBS[OLETE]] | *svid* | (*svid, ...*) | *backup,datum*[*.zeit*] [, FORCE=YES|NO]

Es werden alle Sicherungsversionen, deren Sperrfrist abgelaufen ist, aus der aktuellen Directory-Datei gelöscht. Die Parameter des PURGE-Kommandos sind identisch mit der entsprechenden Archive-Anweisung. Die Parameter DIR= und LIST= werden von CFS automatisch ergänzt.

SORT=*x*[,A|D] Die POOL-Einträge des POOL-Kommandos werden nach dem angegebenen Kriterium sortiert:

C Sortierung nach Spalte "Current Use"
D Sortierung nach Spalte "Device"
E Sortierung nach Spalte "Exp.Date"
O Sortierung nach Spalte "Owner"
R Sortierung nach Spalte "Remark"
V Sortierung nach Spalte "VSN"

A|D Die Sortierung erfolgt in auf- bzw. absteigender Reihenfolge.

/cmd beliebiges BS2000-Kommando

n/OC*n*/CC*n*/DC*n*/CD

beliebiges Connection bezogenes Kommando. Eine vollständige Beschreibung der einzelnen Kommandos finden Sie im Kapitel 9 "Connections" ab Seite 337.

+ | - | +*n* | -*n* | ++ | --

Positionieren innerhalb der Liste

HC *datei* | NHC Ein-/Ausschalten des Hardcopy-Modus zur Protokollierung der Masken

F3 Protokollierung der aktuellen Maske in eine Standard Hardcopy-Datei
Über ein File-Kommando mit LINK=HARDCOPY kann eine Hardcopy-Datei mit einem vorgegebenen Namen zugewiesen werden.

K1 Verlassen der POOL-Maske und Rückkehr in die Dateienliste von CFS

Hinweis:

Mit dem Kommando NPDIR *directory* bzw. DIR *directory* im Feld FILENAME-SELECT der Selektionsmaske werden alle in dem Directory gesicherten Dateien zusammen mit dem letzten Sicherungsstand (Age/Date) in Form einer Dateienliste angezeigt. Für weitere Informationen siehe Seite 53.

Speicherbelegung der Public-Platten anzeigen

PUBSP [*pubset*] [, FILE=*datei*] [, SUM]

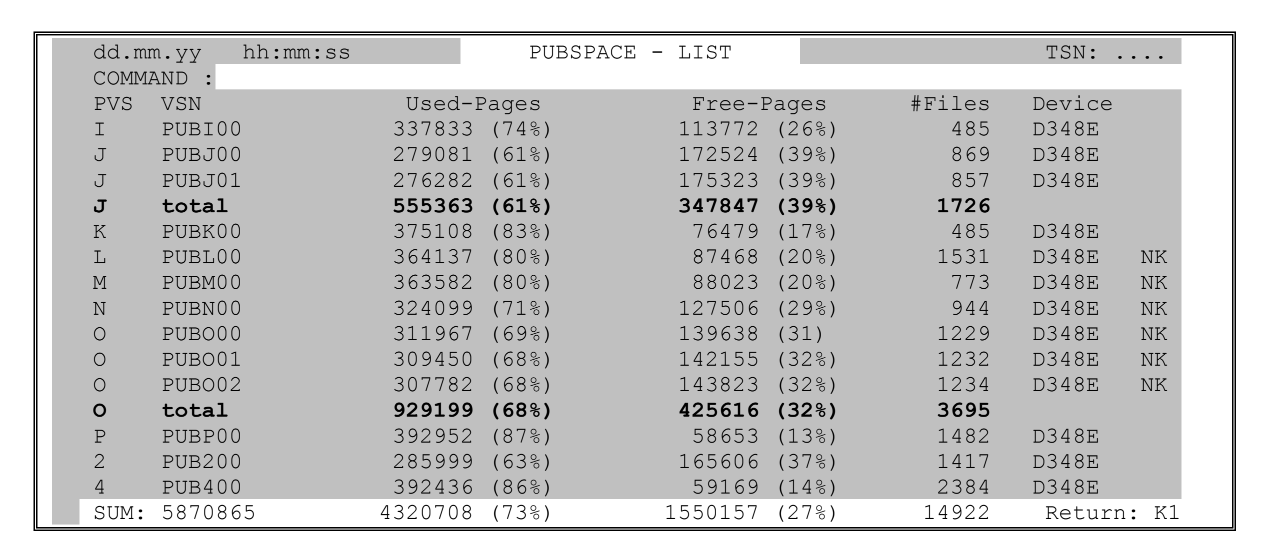
Es wird eine Maske ausgegeben, in der die Speicherbelegung aller Volumes von allen online verfügbaren Pubsets oder eines bestimmten Pubsets angezeigt wird. Voraussetzung für das Kommando PUBSP unter Kennungen ungleich TSOS ist, daß der Systemverwalter unter TSOS über einen Holdertask das Zusatzprodukt TASHT gestartet haben (/ENTER CFS.S. LMSLIB(TASHT)).

pubset Es wird die Speicherbelegung des angegebenen Pubsets angezeigt. Die Pubset-Bezeichnung muß ohne Doppelpunkte angegeben werden, z.B. PUBSP A oder PUBSP TEST.

datei Die Speicherbelegungstabelle wird in einer Datei mit dem angegebenen Namen festgehalten. Die Ausgabe der Bildschirmmaske wird in diesem Fall unterdrückt.

SUM Es wird nur die Summenzeile (siehe unten) für jedes Public-Volume-Set ausgegeben.

Die PUBSP-Maske besitzt folgendes Format:



dd.mm.yy hh:mm:ss PUBSPACE - LIST TSN: ....
COMMAND :

PVS	VSN	Used-Pages	Free-Pages	#Files	Device	
I	PUBI00	337833 (74%)	113772 (26%)	485	D348E	
J	PUBJ00	279081 (61%)	172524 (39%)	869	D348E	
J	PUBJ01	276282 (61%)	175323 (39%)	857	D348E	
J	**total**	**555363 (61%)**	**347847 (39%)**	**1726**		
K	PUBK00	375108 (83%)	76479 (17%)	485	D348E	
L	PUBL00	364137 (80%)	87468 (20%)	1531	D348E	NK
M	PUBM00	363582 (80%)	88023 (20%)	773	D348E	NK
N	PUBN00	324099 (71%)	127506 (29%)	944	D348E	NK
O	PUBO00	311967 (69%)	139638 (31)	1229	D348E	NK
O	PUBO01	309450 (68%)	142155 (32%)	1232	D348E	NK
O	PUBO02	307782 (68%)	143823 (32%)	1234	D348E	NK
O	**total**	**929199 (68%)**	**425616 (32%)**	**3695**		
P	PUBP00	392952 (87%)	58653 (13%)	1482	D348E	
2	PUB200	285999 (63%)	165606 (37%)	1417	D348E	
4	PUB400	392436 (86%)	59169 (14%)	2384	D348E	
SUM:	5870865	4320708 (73%)	1550157 (27%)	14922	Return: K1	

Im Kommandofeld der PUBSP-Maske können folgende Kommandos angegeben werden:

/cmd beliebiges BS2000-Kommando

n/OC*n*/CC*n*/DC*n*/CD
beliebiges Connection bezogenes Kommando. Eine vollständige Beschreibung der einzelnen Kommandos finden Sie im Kapitel 9 "Connections" ab Seite 337.

+ | - | +*n* | -*n* | ++ | --
Positionieren innerhalb der Liste

HC *datei* | NHC Ein-/Ausschalten des Hardcopy-Modus zur Protokollierung der Masken

PUBSP [*pubset*] weiteres PUBSP-Kommando für ein bestimmtes Pubset

NP* Aktualisieren der letzten Pubspace-Liste

LST Verlassen der PUBSP-Maske und Rückkehr in die Dateienliste von CFS bzw. JES (gleiche Wirkung wie K1, siehe unten)

F3 Protokollierung der aktuellen Maske in eine Standard Hardcopy-Datei
Über ein File-Kommando mit LINK=HARDCOPY kann eine Hardcopy-Datei mit einem vorgegebenen Namen zugewiesen werden.

K1 Verlassen der PUBSP-Maske und Rückkehr in die Dateienliste von CFS

Speicherbelegung der Privatplatten anzeigen

PRIVSP [*priv*] [, FILE=*datei*]

Es wird eine Maske ausgegeben, in der die Speicherbelegung aller online verfügbaren Privatplatten angezeigt wird. Voraussetzung für das Kommando PRIVSP unter Kennungen ungleich TSOS ist, daß der Systemverwalter unter TSOS über einen Holdertask das Zusatzprodukt TASHT gestartet haben (/ENTER CFS.S.LMSLIB(TASHT)).

priv — Es wird die Speicherbelegung der Privatplatten angezeigt, deren VSNs mit der angegebenen Teilqualifizierung beginnen. Beispiel: PRIVSP PVT.

datei — Die Speicherbelegungstabelle wird in einer Datei mit dem angegebenen Namen festgehalten. Die Ausgabe der Bildschirmmaske wird in diesem Fall unterdrückt.

Im Kommandofeld der PRIVSP-Maske können folgende Kommandos angegeben werden:

/*cmd* — beliebiges BS2000-Kommando

n/OC*n*/CC*n*/DC*n*/CD
beliebiges Connection bezogenes Kommando. Eine vollständige Beschreibung der einzelnen Kommandos finden Sie im Kapitel 9 "Connections" ab Seite 337.

+ | - | +*n* | -*n* | ++ | --
Positionieren innerhalb der Liste

HC *datei* | NHC — Ein-/Ausschalten des Hardcopy-Modus zur Protokollierung der Masken

PRIVSP [*priv*] — weiteres PRIVSP-Kommando mit einer anderen Teilqualifizierung

NP* — Aktualisieren der letzten Privsp-Liste

LST — Verlassen der PRIVSP-Maske und Rückkehr in die Dateienliste von CFS bzw. JES (gleiche Wirkung wie K1, siehe unten)

F3 — Protokollierung der aktuellen Maske in eine Standard Hardcopy-Datei
Über ein File-Kommando mit LINK=HARDCOPY kann eine Hardcopy-Datei mit einem vorgegebenen Namen zugewiesen werden.

K1 — Verlassen der PRIVSP-Maske und Rückkehr in die Dateienliste von CFS

Vordefinierte P-Tastenbelegung anzeigen

PS [*modul*] [, *maplib*]

P-Show. Anzeigen des Inhalts einer mit dem Softwareprodukt PLUS vordefinierten und in einem Modul gespeicherten P-Tastenbelegung.

Die P-Tastenbelegung kann mit dem Kommando PL geladen werden.

Benutzeranfragen wegen Überschreiben u.ä.

QA | NQA — Query/No Query on Add.
Vor der Ausführung eines ADD in ein bereits existierendes Bibliothekselement wird der Benutzer gefragt, ob er das bereits bestehende Element auch tatsächlich über-

schreiben möchte: ".... is already in library - overwrite ? (Y/N/YA/NA)". Während der Ausführung einer Variablen Action bewirkt die Antwort YA (Yes,All) das Überschreiben des aktuellen Bibliothekselements. Für weitere Elemente werden keine Rückfragen mehr gestellt. Es wird automatisch die Antwort Y angenommen. Analoges gilt für die Antwort NA (No,All). Der YA/NA-Modus wird mit Beendigung der Variablen Action wieder außer Kraft gesetzt.
Standard: NQA (No Query on Add).

QC | NQC
Query/No Query on Copy.
Vor der Ausführung eines Copy in eine bereits existierende Datei/JV/Bibl.-Element wird der Benutzer gefragt, ob er das bestehende Datenobjekt tatsächlich überschrieben werden soll: "... IS ALREADY IN CATALOG - OVERWRITE ? (Y/N/YA/NA)". Während der Ausführung einer Variablen Action bewirkt die Antwort YA (Yes, All) das Überschreiben des Datenobjekts. In weiteren Verlauf werden keine Rückfragen mehr gestellt. Es wird automatisch die Antwort Y angenommen. Analoges gilt für die Antwort NA (No,All). Der YA/NA-Modus wird mit Beendigung der Variablen Action wieder außer Kraft gesetzt.
Standard: NQC (No Query on Copy).

QE | NQE
Query/No Query on Erase.
Um ein unbeabsichtigtes Löschen zu vermeiden, kann der Benutzer mit QE veranlassen, daß jeder Löschvorgang mit dem Action-Code E (Erase) vor der Ausführung eigens bestätigt werden muß. Standard: NQE (No Query on Erase).

Im Zusammenhang mit dem Löschen von Datenobjekten wird auch auf den ERT-Parameter von CFS hingewiesen (ERT = Erase with Retain of Tempfiles).

QR | NQR
Query/No Query on EDT-Return.

QR
Kommando EDT/EDL: Nach der Rückkehr aus dem EDT wird der Benutzer gefragt, ob der virtuelle EDT-Speicher gelöscht werden soll (@D).
Action-Codes EDT/EDL: Nach erfolgreicher Ausführung der Action UPD wird der virtuelle EDT-Speicher automatisch gelöscht.

NQR
Virtueller EDT-Speicher wird nach Action-Code UPD nicht gelöscht; keine Frage an Benutzer wegen @D nach Rückkehr aus dem Kommando-EDT.
Standard: QR.

QS | NQS
Query/No Query on Select.
Vor der Ausführung eines Select auf eine bereits bestehende Datei wird der Benutzer gefragt, ob die Datei tatsächlich überschrieben werden soll: ".... IS ALREADY IN CATALOG - OVERWRITE ? (Y/N/YA/NA)". Während der Ausführung einer Variablen Action bewirkt die Antwort YA (Yes,All) das Überschreiben der Datei. Bei weiteren Dateien werden keine Rückfragen mehr gestellt. Es wird automatisch die Antwort Y angenommen. Analoges gilt für die Antwort NA (No,All). Der YA/NA-Modus wird mit Beendigung der Variablen Action wieder außer Kraft gesetzt.
Standard: QS (Query on Select).

QT | NQT
Query/No Query on Term.
Vor Beendigung des CFS wird der Benutzer gefragt: "DO YOU REALLY WANT TO TERMINATE CFS ? (Y/N)". Standard: NQT.

QL | NQL
Query/No Query on Lock.
Vor dem Sperren des Bildschirms durch das Kommando LOCK wird der Benutzer gefragt: "Do you really want to lock your screen ? (Y/N)". Standard: NQL.

Q | NQ
Query/Non Query.

Q
gleichbedeutend mit QE;QA;QC;QT;QR;QL.

NQ gleichbedeutend mit NQE;NQA;NQC;NQT;NQR;NQL.

Dateien mit RDAC in andere Host-Rechner/PC übertragen/ausdrucken

RDAC [*datei1*] [, *datei2*] [, *passw*] [, *host2*] [, *user-id2*] [, *acct-nr2*] [, *passw2*] [, *rem-succ*] [, *direct*] [, *print*] [, *time*]

Alle Parameter des RDAC-Kommandos sind wahlweise. Falls keine Parameter angegeben sind, so werden diese in der folgenden Maske angefordert.

```
dd.mm.yy   hh:mm:ss       HOST: ........        TSN: ....  USER-ID: ........


                        Transfer-Parameters for RDAC


Filename to be transferred :
Filename in Target-System  : *SAME
Password for remote File   :

Partner-Name                         :
Remote User-ID                       : AAAAAA1
Remote Account-#                     : I0000
Remote Password                      :
Remote Success-Procedure             :


Transfer-Direction         (FROM/TO) : TO
Print ?  (NO/YES/form <,pr-params> ) : NO
Start Time                   (hh:mm) :
```

Parameter des RDAC-Kommandos:

datei1 Name der zu übertragenden Datei.

datei2 Name der Datei im Zielsystem.
Standard: gleicher Name wie *datei1*.

passw Passwort der zu überschreibenden/abzuholenden Datei im Remote-System.
([C]'...' | X'...').

host2 Symbolischer Name des fremden Host-Rechners.
Bei der Angabe von *host2* handelt es sich nicht um den Host-Namen laut PDN-Generierung, sondern um den Partnernamen gemäß dem FT-Netzbeschreibungsbuch im aufrufenden System.

user-id2 User-Id im fremden Host-Rechner, in die die Datei zu übertragen ist bzw. aus der die Datei abzuholen ist. Standard: gleiche Benutzerkennung wie im eigenen Host.

acct-nr2 Abrechnungsnummer.
Standard: gleiche Abrechnungsnummer wie im eigenen Host.

passw2 Logon-Passwort der Benutzerkennung im fremden System ([C]'...' | X'...').
Standard: kein Passwort.

rem-succc Remote Success-Procedure. Name einer Folgeverarbeitung im Remote-System die bei erfolgreicher Übertragung ausgeführt wird.

*proc,** Die als *proc* angegebene DO-Prozedur erhält als Parameter den Namen der übertragenen Datei.

/cmd .... Nach erfolgreicher Übertragung wird im Remote-System das angegebene BS2000-Kommando ausgeführt. Falls '*' im Kommando angegeben wurde, wird an der entsprechenden Stelle der Name der Remote-Datei substituiert. Auf '*' dürfen keine weiteren Parameter für das BS2000-Kommando *cmd* folgen. Falls notwendig, ist hierzu die Variante *proc,** zu verwenden.

'/cmd ...' Nach erfolgreicher Übertragung wird im Remote-System das angegebene Kommando ausgeführt. Das Zeichen '*' wird in dieser Variante nicht durch einen Dateinamen ersetzt.

Standard: keine Remote Success-Procedure.

direct Transfer-Direction FROM | <u>TO</u> (Richtung des File-Transfer).
Standard: Transfer-Direction = TO.

print Soll die Datei zu einem Remote-Drucker übertragen werden ?
<u>NO</u> | YES | *form* [, *pr-params*]

Die Angabe eines Wertes ungleich YES/NO wird als Formularbezeichnung interpretiert und bewirkt einen Ausdruck mit dem angegebenen Form-Parameter.

Im Anschluß an die Werte YES | *form* können wahlweise ein oder mehrere Print-Parameter angegeben werden:

pr-params SPACE= | COPIES= | HEADER= | LINES=nn
Standard: Print = NO.

time *hh:mm* Beginn der Übertragung nicht vor der angegebenen Uhrzeit.
Standard: Beginn der Übertragung zum frühestmöglichen Zeitpunkt.

Hinweise:

Mit dem Kommando RDAC können auch Elemente aus PLAM-Bibliotheken ausgedruckt oder an andere Rechner übertragen werden. Hierzu ist das gewünschte Bibliothekselement mit dem Action-Code * zu markieren und im Kommando anzugeben in der Form RDAC *. CFS selektiert in diesem Fall eine Zwischendatei, die übertragen bzw. ausgedruckt wird.

Werden die Parameter bis einschließlich *acct-nr2* im RDAC-Kommando explizit angegeben, so wird die Parameter Maske nicht ausgegeben und für evtl. noch folgende Parameter werden die Standardwerte angenommen. Um eine Maskenanforderung zu vermeiden, sollte das Kommando RDAC in Prozeduren mindestens mit den folgenden Parametern angegeben werden: RDAC *datei1*, [*datei2*], *host*, *user-id2*, [*acct-nr2*]

Durch Betätigen der K3-Taste kann, ähnlich wie im EDT, ein Bildschirm-Refresh der FT-Maske erreicht werden.

Für weitere Hinweise zum File-Transfer mit RDAC - insbesondere File-Transfer zwischen BS2000 und MS-DOS-Systemen siehe Kapitel 12 "File-Transfer mit openFT/RDAC".

Dateiattribute wiederherstellen

RECAT *datei* [, CATID=*pvs*] [, UID=*user-id*] [, CJ] [,IMP] [,NC] [,NU]

Das Kommando RECAT setzt eine mit Hilfe der Variablen Action ONXLISTCAT *datei* erzeugte Katalogauszugsdatei voraus. Für jeden in der Listcat-Datei gesicherten Katalogeintrag wird geprüft:

a) existiert die Datei ?

b) wenn ja: Stimmen die im Katalogauszug festgehaltenen Dateiattribute wie SHARE=YES/NO, ACCESS=READ/WRITE mit den aktuellen Attributen der datei überein ? Falls Nein, wird ein /CATALOG-Kommando abgesetzt, um die Dateiattribute wieder auf die ursprünglichen Werte zu setzen.

Bei Dateien werden folgende Attribute geprüft und ggf. wieder hergestellt:

- SHARE=YES/NO
- ACCESS=READ/WRITE
- RETPD
- AUDIT
- BACKUP-Level
- LARGE=YES/NO
- DESTROY=YES/NO
- SECONDARY ALLOCATION
- BASIC-ACL
- GUARDS

Bei Jobvariablen werden folgende Attribute geprüft und ggf. wieder hergestellt:

- SHARE=YES/NO
- ACCESS=READ/WRITE
- RETPD
- BASIC-ACL

CATID=*pvs* Die zu bearbeitenden Dateien/Jobvariablen werden auf dem angegebenen Pubset gesucht. Dies erfolgt unabhängig vom der in der Listcat-Datei aufgezeichneten Catalog-ID der Datenobjekte.
pvs ist ohne Doppelpunkte anzugeben, z.B. `CATID=TEST`

UID=*user-id* Die zu bearbeitenden Dateien/Jobvariablen werden unter der angegebenen Benutzerkennung gesucht. Dies erfolgt unabhängig vom der in der Listcat-Datei aufgezeichneten Benutzerkennung der Datenobjekte.
user-id ist ohne $-Zeichen und Punkte anzugeben, z.B. `UID=TT08`

CJ Create Jobvariables. Falls Jobvariablen, deren Katalogeinträge in der Listcat-Datei aufgezeichnet sind nicht mehr existieren, werden sie mit dem ursprünglichen Inhalt und mit den ursprünglichen Attributen wieder angelegt. Falls die Jobvariablen, deren Katalogeinträge in der Listcat-Datei aufgezeichnet wurden, zum Zeitpunkt des RECAT-Kommandos bereits existieren, wird bei der Option CJ der Datenteil mit dem in der Listcat-Datei aufgezeichneten Inhalt gefüllt. Ausnahmen: Die Jobvariable ist ACCESS=READ, hat eine RETPD>0 oder die Jobvariable ist mit einem Paßwort belegt.

IMP Import Files. Falls mit ONXLISTCAT Katalogeinträge von Dateien auf Privatplatten aufgezeichnet wurden und diese zum Zeitpunkt des RECAT-Kommandos nicht mehr im Katalog des Home-PVS existieren (z.B wegen /ER *datei*,CATALOG), wird versucht, diese Dateien wieder in den Katalog Home-PVS zu importieren.

NC — No Cat-ID's. Die in der LISTCAT-Datei enthaltenen CAT-ID's der Katalogeinträge werden bei der Feststellung der aktuellen Dateiattribute nicht berücksichtigt. Die zu bearbeitenden Dateien/Jobvariablen werden auf dem Default-Katalog der jeweiligen Kennung gesucht.

NU — No User-ID's. Die in der LISTCAT-Datei enthaltenen User-ID's der Katalogeinträge werden bei der Feststellung der aktuellen Dateiattribute nicht berücksichtigt. Die zu bearbeitenden Dateien/Jobvariablen werden in der aktuellen Benutzerkennung gesucht.

Hinweise:

Der CJ-Parameter des RECAT-Kommandos kann nach erfolgreichem File-Transfer einer Listcat-Datei zum Installieren von Jobvariablen auf einem fremden Host verwendet werden. Siehe hierzu Seite 417.

Die RECAT-Funktion kann auch als Stand-Alone Modul aus der CFSLIB aufgerufen werden: /EXEC (RECAT,CFSLIB). Die als Parameter benötigten Operanden *datei* [,P] [,CJ] [,IMP] werden dann über SYSDTA angefordert.

Bibliothek / ZIP-Archiv / Datei reorganisieren

REORG *bibl-1* [, *bibl-2*] [, *par*]

Die Bibliothek *bibl-1* (FMS/LMS/PLAM), das ZIP-Archiv, bzw. Datei wird reorganisiert, indem alle Elemente (Datensätze) in eine neu angelegte Hilfsdatei *bibl-2* übertragen werden.
Durch das Kommando REORG werden die durch häufiges Löschen und Neuaufnehmen von Elementen entstandenen "Löcher" aus der Bibliothek entfernt. Die Bibliothek beansprucht am Ende weniger Speicherplatz als vor der Reorganisation.

par — [*vsn/device*] | [*:x:*] [,*smspar*] [, PAD=*nn*] [, NR] [, DF[O]] [, SPACE=(*xxxx,yyy*)] [, MODE=CFS|LMS]

vsn/device — Die reorganisierte Bibliothek/die zur Reorganisation benötigte Hilfsdatei wird auf der angegebenen Privatplatte eingerichtet.

Hinweis:

Falls sich die Hilfsdatei über mehrere Privatplatten erstrecken soll oder falls sie auf einem anderen speziellen Speichermedium eingerichtet werden soll, so kann dies auf die folgende Weise erreicht werden:
Der Benutzer legt auf den gewünschten Datenträgern eine Datei mit beliebigem Namen (Ausnahme: CFS.REORGFILE) und in ausreichender Größe an. Diese Datei wird CFS durch das Kommando /FILE *datei*,LINK=CFSREORG bekannt gemacht. In der Folge werden alle Reorganisationen mit dieser Datei als Hilfsdatei durchgeführt. Die Hilfsdatei wird nach Abschluß der Reorganisation nicht gelöscht, wie dies normalerweise der Fall ist. Es wird lediglich ein Kommando /ERASE *datei*,DATA zum Löschen der Dateninhalte ausgeführt. Der reservierte Speicherplatz bleibt hierdurch bestehen.

:x: — Die reorganisierte Bibliothek/die zur Reorganisation benötigte Hilfsdatei wird auf dem angegebenen PVS eingerichtet.

smspar — Falls die reorganisierte Bibliothek/die zur Reorganisation benötigte Hilfsdatei auf einem SM-Pubset (System Managed Pubset) angelegt wird, so kann mit diesem Parameter in Abhängigkeit von den Vorgaben des Systemverwalters der Ablageort (Volume-Set) der Datei genauer bestimmt werden.

AVAILABILITY=HIGH
Die Datei soll innerhalb eines SM-Pubsets auf einem Volume-Set abgelegt werden, der erhöhte Ausfallsicherheit besitzt (z.B. durch DRV-gespiegelte Platten).

STORAGE-CL=*class*
Name einer vom Systemverwalter eingerichteten Storage Klasse. Über die Storage Klasse wird das Volume-Set und damit die Speichereigenschaften der Datei (z.B. Performance Attribute, Availability) festgelegt.

VOLUME-SET=*vset*
Bestimmt das Volume-Set, auf dem die Datei innerhalb des SM-Pubsets eingerichtet werden soll. Der Systemverwalter kann mit *CONTROL bzw. mit der Angabe einer ein- bis vierstelligen Cat-Id *vset* (ohne ':') einen Volume-Set explizit angeben.
Der nichtprivilegierte Benutzer kann einen Volume-Set nur dann angeben, wenn ihm der Systemverwalter das Recht der physikalischen Allokierung erteilt hat.

WORK-FILE=YES|NO
Gibt an, ob die Datei eine Arbeitsdatei ist, die von der Systemverwaltung zu einem bestimmten Zeitpunkt gelöscht werden darf. Work-Dateien werden auf einem Volume-Set abgelegt, für das der Systemverwalter die Nutzungsart WORK vorgegeben hat.

PAD=*nn*
Optionaler PAD-Faktor für die reorganisierte Datei. Diese Angabe wird nur bei der Reorganisation von ISAM-Dateien berücksichtigt. PAD=*nn* kann auch nach *:x:* bzw. *vsn/device* angegeben werden (getrennt durch Komma).
Standard: PAD=0

NR
No Release of free Space. Zugewiesener, jedoch nicht beschriebener Speicherplatz wird nach Abschluß der Reorganisation nicht freigegeben. Dieser Speicherplatz steht der Datei/Bibliothek für spätere Erweiterungen zur Verfügung.
Standard: Der nicht benötigte Speicherplatz wird freigegeben.

DF[O]
Defragment [Only]. Vor dem Zurückkopieren der reorganisierten Hilfsdatei wird der Speicherplatz der Originaldatei mit /ERASE ,SPACE freigegeben und in der benötigten Größe neu angefordert. Es läßt sich damit die Anzahl der Extents der Originaldatei verringern. Die Defragmentierung ist besonders effizient, wenn die Hilfsdatei auf einem anderen Pubset angelegt wurde (REORG *bibl*,*:x:*,DF).
DFO: Es wird nur die Defragmentierung durchgeführt. "Löcher" in Bibliotheken/ISAM-Dateien werden hierbei nicht freigegeben.
Standard: Defragmentierung ausgeschaltet bei Bibliotheken und ISAM-Dateien. Defragmentierung eingeschaltet bei SAM-Dateien und PAM-Dateien, die keine Bibliotheken sind.

SPACE=
Nach erfolgter Reorganisation wird ein FILE-Kommando mit dem angegebenen SPACE-Parameter ausgeführt. Es kann damit überschüssiger Speicherplatz freigegeben und gleichzeitig die Sekundärzuweisung für die Datei modifiziert werden (z.B. SPACE=(-9999,30)).
Standard: Der nicht benötigte Speicherplatz wird freigegeben. Die Sekundärzuweisung wird nicht verändert.

MODE=CFS|LMS
: Dieser Parameter legt die angewendete Methode der Reorganisation von PLAM-Bibliotheken fest.

MODE=CFS
: CFS-eigene, gründliche Reorganisation. Hierbei werden alle Elemente in eine neue Bibliothek übertragen. An Ende wird die neu erzeugte Hilfsbibliothek auf die zu reorganisierende Bibliothek zurückkopiert und danach gelöscht. Diese Methode bringt optimale Reorganisationsergebnisse, nimmt aber besonders bei großen Bibliotheken deutlich mehr Zeit in Anspruch als die LMS-Methode (s.u.).

MODE=LMS
: LMS bietet ab Version 3.0 eine eigenständige Reorganisationsmethode an. Diese arbeitet inplace, d.h. ohne die einzelnen Elemente zu übertragen und am Ende die Bibliothek zu kopieren. Mit dieser Methode wird eine Bibliothek in kürzestmöglicher Zeit reorganisiert. Unter Umständen werden weniger Seiten freigegeben als bei der CFS-Reorganisation.
Standard: MODE=CFS.
Über CFSMAIN kann der Standard aber auch auf MODE=LMS eingestellt werden.

Hinweise:

Falls *bibl-2* nicht angegeben, wird die im Rahmen der Reorganisation erzeugte Hilfsdatei auf die ursprüngliche Bibliothek *bibl-1* zurückkopiert und mit dem Kommando /FILE ...,SPACE=-9999 nicht mehr benötigter Speicherplatz freigegeben. Die Hilfsdatei wird anschließend gelöscht.

Falls *bibl-2* angegeben wurde, so wird die reorganisierte Bibliothek/ISAM-Datei unter diesem Namen angelegt und am Ende nicht auf die Originalbibliothek zurückkopiert. Die Ausgabedatei *bibl-2* ist in diesem Fall auch bezüglich der benötigten Extents optimiert.

Falls in der Dateienliste Elemente einer Bibliothek angezeigt werden, so kann im Kommando REORG der Name der zu reorganisierenden Bibliothek (*bibl-1*) weggelassen werden. In diesem Fall wird die Bibliothek reorganisiert, deren Elemente in der Dateienliste angezeigt sind.

Beispiel:

```
REORG MACROLIB,:TEST:,DF
```
Die PLAM-Bibliothek MACROLIB wird reorganisiert und defragmentiert. Die für die Reorganisation notwendige Hilfsdatei wird auf dem Pubset TEST angelegt. Am Ende der Reorganisation wird der Speicherplatz der Originaldatei freigegeben und danach in der benötigten Größe wieder neu angefordert. Die Hilfsdatei auf dem PVS :TEST: wird dann auf die Originaldatei zurückkopiert und auf :TEST: gelöscht.

Kontrolle an CFS-Prozedur zurückgeben

RET

Die folgenden Eingaben für das Programm CFS werden von der Eingabequelle SYSDTA angefordert. Falls CFS mit einer STARTUP-Datei hochgefahren wurde, so bewirkt RET, daß die weiteren Eingaben aus der Startup-Datei gelesen werden.

Das Kommando RET setzt voraus, daß in der Kommandodatei/Startup-Datei durch die CFS-Prozeduranweisung *DIA die Kontrolle irgendwann von der Prozedur auf den Benutzer am Bildschirm übergeben wurde. Will der Benutzer den prozedurgesteuerten Ablauf wieder aufnehmen, so ist dies über das Kommando RET möglich.

RET *datei* | RET *bibl* (*element*)

Durch eine Parameterangabe beim RET-Kommando wird vor dem Umschalten in den Prozedurmodus eine neue Eingabequelle für SYSDTA zugewiesen:
SYSFILE SYSDTA=*datei* | SYSFILE SYSDTA=*bibl (element)*.

Hinweise:

Weitere Informationen zum Thema Ablauf von CFS in Prozeduren finden Sie auf Seite **485** und folgende (Prozedursprache).

Die Prozedursprache von CFS bietet auch Syntaxelemente für eine automatische, an bestimmte Bedingungen (z.B. Benutzereingaben) geknüpfte Wiederaufnahme der Prozedur. Nähere Informationen hierzu finden Sie auf Seite **505**.

Rewrite-Kommando für mehrfachen Update

Mit der Variablen Action ONXFIND... =W *datei* / User Option FIND ... =W *datei* kann eine druckaufbereitete Datei erzeugt werden, in der alle Sätze der durchsuchten Datenobjekte aufgelistet sind, die einen bestimmten String (z.B. 'D3480') enthalten. Die Namen der Datenobjekte sind ebenfalls in der Write-Datei enthalten. In einer Vielzahl von Anwendungsfällen ist es nun vorteilhaft, in den gesammelten Datensätzen der Write-Datei zentral Änderungen vorzunehmen (z.B. 'D3480' durch 'D3490' zu ersetzen) und diese Änderungen mit einem Kommando in die Datenobjekte zurückzuschreiben.

Diese Funktion leistet das Rewrite-Kommando von CFS. In der Write-Datei können sogar Sätze von Dateien, Bibliothekselementen und Jobvariablen gemischt vorkommen (FIND...=W *datei*,E). Ein Beispiel für die Anwendung des Rewrite-Kommandos finden Sie auf Seite **535**.

Wirkungsweise des Rewrite-Kommandos

In der Write-Datei wird ein Header-Satz der Art: FILE= / ELEMENT=/ JV= gesucht und das entsprechende Datenobjekt im Input-Modus eröffnet. Ein zweites Datenobjekt wird im Output-Modus eröffnet. Sodann werden die im Eingabedatenobjekt und in der Write-Datei enthaltenen Datensätze gelesen und in das Ausgabedatenobjekt übertragen, wobei Sätze aus der Write-Datei die entsprechenden Sätze aus dem Eingabeobjekt überschreiben. Nachdem alle Sätze verarbeitet wurden, wird das neu erzeugte Ausgabeobjekt auf das Original zurückkopiert und anschließend gelöscht (Standard). Aufgrund zusätzlicher Optionen kann das Original auch unverändert beibehalten werden.

Treten während der Bearbeitung eines Datenobjekts Fehler auf, so wird ein Fehlerprotokoll erstellt. Dieses erhält den Namen der Write-Datei, wobei das Suffix '.ERR' angehängt wird (*datei*.ERR).

Syntax des Rewrite-Kommandos

REWR *datei* [, N] [, OLD=*suffix-1*] [, NEW=*suffix-2*] [, AI] [, NCD] [, ARO] [, XL] [, CL=*nn*]

datei Name der Write-Datei mit den zu modifizierenden Datensätzen und den Namen der zugehörigen Datenobjekte. Diese Datei muß mit der Variablen Action ONXFIND... =W bzw. mit der User Option FIND...=W erzeugt worden sein. Als Eingabedatei kann auch die durch ein früheres Rewrite-Kommando erzeugte Fehlerdatei dienen. Diese Fehlerdatei enthält die Datensätze zu allen Objekten, die in dem früheren Rewrite-Lauf nicht zurückgeschrieben werden konnten.

N Names-Option.
Während des Rewrite werden die Namen der gerade bearbeiteten Datenobjekte am Bildschirm angezeigt. Standardmäßig werden die Namen nicht angezeigt. Lediglich

die Namen der Datenobjekte, für die der Rewrite aufgrund irgendwelcher Fehler nicht ausgeführt werden konnte, werden zusammen mit den entsprechenden Fehlermeldungen am Bildschirm ausgegeben.

OLD=*suffix-1*

Originale unter dem angegebenen Suffix sichern.
Vor Ausführung des Rewrite wird eine Sicherungskopie des Originals angelegt.

Rewrite von Dateien/Jobvariablen: Nach erfolgreicher Ausführung des Rewrite stehen die unveränderten Originale unter dem Namen *xxx.suffix-1* zur Verfügung. *xxx* ist der Name der modifizierten Datei/Jobvariablen.
Beispiel: OLD=.ALT Nach Durchführung der Änderungen stehen die unveränderten Originaldateien noch unter dem Namen *xxx*.ALT zur Verfügung.

Rewrite von LMS-/PLAM-Bibliothekselementen ohne Parameter XL:
Das unveränderte Element wird in der gleichen Bibliothek unter der Versionsbezeichnung *suffix-1* sichergestellt. Beginnt *suffix-1* mit einem Punkt, so wird dieser für die Versionsbezeichnung ignoriert.
Beispiel: OLD=.ALT Nach Durchführung der Änderungen stehen die unveränderten Originalelemente noch unter dem Namen [*t*/]*xxx* /ALT zur Verfügung.

Rewrite von FMS-Bibliothekselementen ohne Parameter XL:
Nach erfolgreichem Rewrite stehen die unveränderten Originalelemente in der gleichen Bibliothek unter dem Namen *xxx.suffix-1* zur Verfügung. *xxx* ist der Name des jeweils zu modifizierenden FMS-Elements.

Rewrite von FMS-/LMS-/PLAM-Bibliothekselementen mit Parameter XL:
Die unveränderten Elemente werden unter dem gleichen Namen in einer separaten Bibliothek *xxx.suffix-1* gesichert. *xxx* ist hierbei der ursprüngliche Bibliotheksname.
Beispiel: OLD=.ALT Vor der Durchführung der Änderungen werden die betroffenen Elemente unter dem originalen Namen in einer Bibliothek mit dem Suffix .ALT gesichert.

NEW=*suffix-2*

Modifizierte Datenobjekte unter dem angegebenen Suffix ablegen.

Rewrite von Dateien/Jobvariablen: Nach erfolgreichem Rewrite stehen die modifizierten Objekte unter dem Namen *xxx.suffix-2* zur Verfügung. *xxx* ist der Name der unveränderten Originaldatei/Jobvariablen.
Beispiel: NEW=.REWR Nach Durchführung der Änderungen stehen die geänderten Dateien unter dem Namen *xxx*.REWR zur Verfügung.

Rewrite von LMS-/PLAM-Bibliothekselementen ohne Parameter XL:
Die geänderten Elemente erhalten die Versionsbezeichnung *suffix-2*. Beginnt *suffix-1* mit einem Punkt, so wird dieser für die Versionsbezeichnung ignoriert.
Beispiel: NEW=.REWR Nach Durchführung der Änderungen stehen die geänderten Elemente unter dem Namen [*t*/]*xxx*/REWR zur Verfügung. Bei LMS-ISAM Bibliotheken (OSM-Format) wird die Versionsbezeichnung auf REW verkürzt, da in diesem Fall nur dreistellige Versionsbezeichnungen möglich sind.

Rewrite von FMS-Bibliothekselementen ohne Parameter XL:
Nach erfolgreichem Rewrite stehen die geänderten Elemente unter dem Namen *xxx.suffix-2* zur Verfügung.

Rewrite von FMS-/LMS-/PLAM-Bibliothekselementen mit Parameter XL:
Die veränderten Elemente werden unter dem gleichen Namen in eine separate Bibliothek *xxx.suffix-2* geschrieben. *xxx* ist hierbei der ursprüngliche Bibliotheksname.
Beispiel: NEW=.NEU Die geänderten Elemente werden unter dem originalen Namen in einer in einer eigenen Bibliothek mit dem Suffix .NEU abgelegt.

AI — Add with Increment of version.

Während der Ausführung des Rewrite wird temporär der AI-Modus von CFS eingeschaltet. Beim Rewrite von Bibliothekselementen (LMS-ISAM/PLAM) wird damit bewirkt, daß die Versionsbezeichnungen der neu erzeugten Elemente gegenüber den Ursprungselementen um 1 erhöht werden.

NCD — No Check for Last Update.

Beim Wegschreiben der Treffersätze aus einer BS2000-Datei unter OSD V1 oder höher bzw. aus einer PLAM-Bibliothek wird neben dem Namen des beteffenden Datenobjekts auch dessen Last Update Datum in der Trefferdatei abgelegt. Beim Rewrite wird in diesem Fall geprüft, ob das Datenobjekt seit dem ONXFIND-Lauf geändert wurde. Ist dies der Fall, wird der Rewrite für das Datenobjekt nicht ausgeführt, da dies inkonsistente Daten zur Folge haben kann.
Mit der Option NCD wird die Last Update Prüfung durch den Benutzer ausgeschaltet.

ARO — Access Read Overwrite.

Mit dieser Option können auch Dateien zurückgeschrieben werden, die die Attribute ACCESS=READ und/oder RETPD>0 besitzen.

XL — eXtra Library.

Dieser Parameter hat nur eine Wirkung beim Rewrite von Bibliothekselementen und falls der Parameter OLD=*suffix-1* oder NEW=*suffix-2* angegeben wurde. Die unveränderten (OLD=) bzw. veränderten (NEW=) Bibliothekselemente werden, abweichend vom Standardverfahren, unter dem originalen Namen in eigene Bibliotheken *name.suffix-1/2* geschrieben. *name* ist dabei jeweils der Name der gerade bearbeiteten Bibliothek.

CL=*nn* — Check Length.

Die in der FIND-Datei enthaltenen Treffersätze werden vor dem Zurückschreiben in die Original Datenobjekte auf ihre Länge geprüft. Bei Überschreitung der angegebenen Satzlänge wird das entsprechende Datenobjekt nicht zurückgeschrieben. Die fehlerhaften Sätze der Write-Datei werden in die Error-Datei übertragen (siehe unten: "Fehlerbehandlung bei Rewrite").
Der angegebene Wert *nn* entspricht der maximalen Länge, die die Sätze aus der Write-Datei nach dem Zurückschreiben in das Original-Datenobjekt haben dürfen. Das Satzlängenfeld (4 Byte) wird hierbei nicht mitgerechnet.
Beispiel: `REWR datei,CL=72`
Durch das Ändern von Strings in der Write-Datei mit dem EDT (z.B. Anweisung `ON&C'T9G'T'TAPE-C4'`) kann es vorkommen, daß Datensätze zu lang werden, so daß bei DO-Prozeduren nach dem Rewrite das Fortsetzungszeichen '-' weiter als bis Spalte 72 hinaus geschoben wird. Durch CL=72 werden solche Datenobjekte nicht zurückgeschrieben sondern in die Error-Datei übernommen.

Hinweise:

Funktion FIND mit Wegschreiben der Treffer (ONXFIND...=W *datei*): Bei Prozeduren /Makros/Quellprogrammen werden alle Sätze als logische Einheit betrachtet, die durch Fortsetzungszeichen miteinander verbunden sind. Falls in einem dieser Sätze das Suchitem gefunden wurde, so werden alle Sätze der logischen Einheit in die Write-Datei übertragen. Der Satz, der den Treffer enthält, wird durch das Zeichen 'X' in Spalte 2 hervorgehoben.

Bei PLAM-Bibliothekselementen mit der Default-Versionsbezeichnung X'FF' bewirkt das Zurückschreiben mit der AI-Option, daß das Ursprungselement die Versionsnummer 001 erhält und das neu erzeugte Element die Versionsnummer 002. Die AI-Option ist bei Rewrite von FMS-Elementen ohne Wirkung.

Die Bearbeitung der Write-Datei kann im CFS-Editor (Action-Code M (Modify)) bzw. im EDT vorgenommen werden.
Die Write-Datei kann Datensätze mit einer Länge von bis zu 2040 Bytes enthalten (PAM-Blöcke, Format-B Sätze aus PLAM-Bibliothekselementen).

Vor dem Modifizieren der Write-Datei sollte die Originaldatei in einer Kopie sichergestellt werden.

Fehlerbehandlung bei Rewrite

Tritt beim Rewrite für ein Datenobjekt ein Fehler auf, so werden alle zugehörigen Sätze aus der Eingabedatei (REWR *datei*) in eine Fehlerdatei mit dem Namen *datei*.ERR übertragen. Danach wird mit dem Rewrite für das nächste Datenobjekt fortgefahren. Nachdem der Benutzer die Fehlerursache geklärt hat, kann ausschließlich für die zuvor nicht erfolgreich zurückgeschriebenen Elemente ein erneuter Rewrite durchgeführt werden: REWR *datei*.ERR

Ein häufiger Fehler beim Bearbeiten der Rewrite-Datei besteht darin, daß mit dem EDT-Kommando @ON&C'...'T'...' auch Änderungen in Sätzen der Rewrite-Datei vorgenommen werden, die die Namen der Datenobjekte enthalten. Um versehentliche Änderungen in diesen Sätzen zu vermeiden, wird vorgeschlagen, alle Sätze mit Kontrollinformationen für Rewrite zu Beginn der EDT-Bearbeitung in eine zweite Ebene zu kopieren (ON&:2-2:FIND'*'COPY TO (1) KEEP). Nach erfolgter Bearbeitung der Datensätze können die Sätze mit Kontrollinformationen wieder an ihre Ursprungsposition zurückkopiert werden (COPY &(1)).

Kann eine von Rewrite erzeugte Hilfsdatei aus irgendwelchen Gründen nicht auf die Ursprungsdatei zurückkopiert werden, so wird eine entsprechende Meldung in das Rewrite-Fehlerprotokoll geschrieben und die Hilfsdatei (Name = *eingabedatei*.REWR) wird nicht gelöscht.

Editieroptionen bei Rewrite

Die Rewrite-Funktion von CFS kann auch dazu benutzt werden, neue Datensätze in die Objekte einzubringen bzw. vorhandene Datensätze zu löschen.

Für diese erweiterten Editieroptionen ist das Format der durch ONXFIND erzeugten Write-Datei zu beachten.

Die Datensätze der Write-Datei haben folgendes Format:

```
DNNNNNNNX................
|  |      |  |
|  |      |  ▫-- Dateninhalt (evtl. mit Zusatzinformationen wie PAM-Key).
|  |      |      Satzart bzw. Record-ID bei Sätzen in PLAM Bibliotheken)
|  |      |
|  |      ▫----- X: Markierungsspalte (Standardmäßig Blank)
|  |
|  ▫------------ N..N: Satz-/Blocknummer
|
▫--------------- D: Drucksteuerzeichen (SPACE=E)
```

Einfügen von Datensätzen

Falls nach einem in der Write-Datei aufgeführten Satz, *n* zusätzliche Sätze eingefügt werden sollen, so sind diese in der Write-Datei im Anschluß an den Satz mit der Nummer *n* einzutragen. Bei den eingefügten Sätzen sind die Spalten DNNNNNNX leer zu lassen (fortlaufende Einfügung). Der Dateninhalt der einzufügenden Sätze beginnt in Spalte 10. Es können beliebig viele Sätze eingefügt werden.

Falls nach einem in der Write-Datei nicht aufgeführten Satz neue Sätze eingefügt werden sollen, sind diese nach dem in der Write-Datei enthaltenen Satz mit der nächst kleineren Satznummer einzufügen. Beim ersten eingefügten Satz ist in der Satznummernspalte N..N die Satznummer *n* einzutragen, falls nach einem bestehenden Satz *n* eingefügt werden soll. In der Markierungsspalte X ist das Zeichen 'A' einzutragen (A =insert After). Auf die Markierungsspalte X folgt der einzufügende Dateninhalt. Sind 2 oder mehr fortlaufende Sätze einzufügen, so sind für den zweiten und alle folgenden Sätze die Spalten N..N und X auf Blank zu setzen (fortlaufende Einfügung). Falls vor dem ersten im Datenobjekt enthaltenen Satz eingefügt werden soll, so ist bei N..N die Satznummer 0 einzutragen. Soll nach dem letzten im Datenobjekt enthaltenen Satz eingefügt werden, so ist bei N..N eine hypothetisch große Satznummer anzugeben, z.B. 9999999.

Ersetzen von Datensätzen

Falls ein Datensatz vollständig durch einen neuen ersetzt werden soll (Replace), so ist bei N..N die Nummer des gewünschten Satzes anzugeben. Die Markierungsspalte X ist auf Blank zu setzen. Das Ersetzen von bereits vorhandenen Datensätzen ist die Standardfunktion des Rewrite-Kommandos.

Löschen von Datensätzen

Durch Eintragen des Zeichens 'E' (Eliminate) oder 'D' (Delete) in der Markierungsspalte X wird der Satz/PAM-Block mit der in N..N angegebenen Nummer aus dem Original gelöscht. Hierbei ist es nicht notwendig, daß im Datenteil der Inhalt des Datensatzes aufgeführt ist. Durch Einfügen eines Satzes von der Art '⊔ ⊔ ⊔ ⊔ ⊔ ⊔78E' wird beispielsweise der 78. Satz gelöscht.

Hinweis: Durch die Variable Action ONXFIND ''=W *datei* ist es möglich, alle Sätze eines Datenobjekts in die Write-Datei zu schreiben.

Gespeicherte Dateienliste wieder aktivieren

RL [*mn*] [*,sel*] Restore List. Es wird eine früher vom Benutzer selektierte Datei- bzw. Bibliotheksauswahl als Dateienliste wieder angezeigt. Die Anzeige beginnt mit dem in der früheren Dateienliste zuletzt angezeigten Ausschnitt.

mn beliebige ein- bis achtstellige alphanumerische Kurzkennung (Mnemo-Code), unter der die zu aktivierende Dateienliste mit dem Kommando SL (Save List) gespeichert wurde (SL *mn*).

Auch ohne ein explizites SL-Kommando durch den Benutzer kann mit
RLF die letzte Liste mit BS2000-Dateien
RLJ die letzte Liste mit Jobvariablen
RLL die letzte Liste mit Bibliothekselementen
RLD die letzte Liste mit Archive-Directory-Einträgen
RLZ die letzte Liste mit Elementen eines ZIP-Archivs

wieder hergestellt werden, da diese Listen von CFS unter den entsprechenden Namen automatisch gesichert werden.

Falls *mn* nicht angegeben, so wird die vorletzte Dateienliste wieder aktiviert. Die vorletzte Dateienliste ist diejenige Dateienliste, die der derzeit aktuellen voraus ging.

sel Es werden nur die Einträge angezeigt, deren Namen zusätzlich der Selektionsbedingung *sel* genügt.
Beispiel: RLL,TMP.
Alle Einträge der letzten LMS-Liste, die zusätzlich auch noch die Zeichenfolge TMP. im Namen enthalten.

Das Kommando RL ohne Angabe des Mnemo-Codes einer mit SL gesicherten Liste ist nur möglich, falls der ADOC-Modus von CFS eingeschaltet ist (Standard).

Zum Thema "frühere Dateienlisten aktivieren" siehe auch Kommando SL (Save List) Seite **281**, ADOC (Automatic Documentation) Seite **468** und DOC Seite **473**, sowie Feld DOCUMENTATION in der Selektionsmaske Seite **74**.

Beispiel:

Parallele Bearbeitung mehrerer Bibliotheks- und Dateienlisten.
Der Benutzer hat als Dateienliste eine Menge von Bibliotheken selektiert. Mit dem Action-Code NP kann er zur Anzeige des Inhaltsverzeichnisses einer der dargestellten Bibliotheken verzweigen. Das Inhaltsverzeichnis kann z.B. mit SL 1 gesichert werden. Mit dem Kommando RL gelangt er wieder in die Liste der Bibliotheken. Hier kann mit dem Action-Code NP das Inhaltsverzeichnis der nächsten Bibliothek aufgeblättert und mit SL 2 gesichert werden. Dieser Vorgang kann für alle Bibliotheken wiederholt werden. Ein direktes hin- und herspringen zwischen den Bibliotheken ist über die Kommandos RL 1, RL 2 usw. möglich.

Eintrag in Dateienliste suchen

S [-],'*string*' In der Dateienliste wird das nächste Vorkommen der angegebenen Zeichenfolge gesucht. Der angezeigte Ausschnitt wird so positioniert, daß der Treffer in der ersten angezeigten Zeile steht.
Für eine Beschreibung der vollen Syntax des Suche-Kommandos wird auf die Abschnitte "Suchen von Zeichenfolgen (einfaches Suchargument / mehrere Suchargumente)" auf Seite **309** und folgende verwiesen.

\- Rückwärtssuche: Es wird von der ersten am Bildschirm angezeigten Zeile der Dateienliste in Richtung Anfang gesucht.
Standard (ohne Angabe von '-'): Es wird von der ersten am Bildschirm angezeigten Zeile der Dateienliste in Richtung Ende gesucht.

S [-] In der Dateienliste das nächste Vorkommen der im letzten Suche-Kommando definierten Zeichenfolge suchen. Der wahlweise Zusatz '-' bewirkt eine Suche in Richtung Anfang der Dateienliste.

S? Das zuletzt definierte Suchargument anzeigen.

Hinweis:

Mit dem Kommando SC, siehe unten, kann ein Zeilenlineal in die Dateienliste eingeblendet werden. Damit erhält der Benutzer eine Orientierung, falls er das Suche-Kommando auf einen Spaltenbereich innerhalb der Dateienliste anwenden möchte.

Beispiel:

`S,:11:'SRC.'`
Suche in der angezeigten Liste eine Datei, deren Name mit 'SRC.' beginnt.

Suchen in Dateienliste mit direkter Ausgabe der gefundenen Einträge am Bildschirm

S,'*string*'**=P** Die angegebene Zeichenfolge wird in der Dateienliste von der aktuellen Position bis zum Ende gesucht. Einträge, die Zeichenfolge enthalten, werden im Line-Modus am Bildschirm ausgegeben und die Suche wird fortgesetzt. Diese Form des Suche-Kommandos entspricht der Anweisung ON&P'....' des EDT.

Nach Ausgabe von maximal 21 Sätzen wird eine Überlaufkontrolle eingefügt ("Please acknowledge / End: K1"). Durch Betätigung der K1-Taste wird der Suchvorgang beendet. Jede andere Eingabe bewirkt eine Fortsetzung der Suche.

Nach Beendigung des Suchvorgangs wird die Dateienliste wieder ab der Position angezeigt, bei der das Suche-Kommando eingegeben wurde.

Hinweis:

Nach Beendigung der Anzeige der Treffer ist die Suchbedingung noch gespeichert. Durch Eingabe des Kommandos S kann das Sichtfenster zum ersten Eintrag positioniert werden, der das Suchitem enthält.

Suchen in Dateienliste und bei Treffern Action-Code eintragen

S,'*string*'**=IN[SRT]** *act*

Die angegebene Zeichenfolge wird in der Dateienliste von der aktuellen Position bis zum Ende gesucht. Bei Einträgen, die diese Zeichenfolge enthalten, wird der angegebene Action-Code *act* eingetragen. Die Suche wird bis zum Ende der Dateienliste fortgesetzt.

Anstelle eines einfachen Sucharguments '*string*' ist auch Kombination aus mehreren Suchargumenten mit den Verknüpfungszeichen */,/+ möglich.

Beispiel:

`S,:11:-'TAS'=INSRT-`
Streiche aus der angezeigten Liste alle Bibliothekselemente, deren Namen nicht mit der Zeichenfolge 'TAS' beginnt.

Lineal einblenden

SC | <u>SCO</u> SCale/SCale Off

SC In der oberen Bildschirmhälfte wird ein "Lineal" mit einem Spaltenzähler angezeigt. Das Lineal wird im Display-Modus und in der Dateienliste eingeblendet.
Alternativ zu SCO kann auch das Kommando NSC verwendet werden.

OL | NOL Anstelle SC/SCO/NSC sind auch die Kommandos OL/NOL (Orientation Line/ No Orientation Line) zulässig.

Standard: SCO (kein Lineal anzeigen).

PLAM-Bibliotheken: Age aus "System-Date" bestimmen

SDAT PLAM-Bibliothekselemente: Die Age-Angaben in der Elementliste werden aufgrund des vom Benutzer nicht beeinflußbaren System-Datums (=Datum des letzten Updates) errechnet. Standard: UDAT

Hinweis:

Beim Reorganisieren von PLAM-Bibliotheken wird das System-Datum aller Elemente auf den aktuellen Tag gesetzt. Das User-Datum bleibt unverändert. Gleiches gilt auch, falls mit LMS REP-Anweisungen in ein Modul vom Typ R eingebracht werden.

Datei aus Bibliothekselement erzeugen (Select)

SEL [*t/*] *element, bibliothek* [, *datei*] [, *act*]

Aus dem angegebenen Bibliothekselement soll eine Datei erzeugt werden. Den Namen der neuen Datei bestimmt der Benutzer mit dem Operanden *datei*. Fehlt dieser, so sind Elementname und Dateiname identisch.

t/ Bei der Selektion eines Elements aus einer PLAM-Bibliothek sollte der Elementtyp *t/* angegeben werden. Wurde der Elementtyp nicht angegeben, so wird dieser von CFS nachgefordert.

Falls die Dateienliste aus Dateien besteht, so wird die mit dem SEL-Kommando neu erzeugte Datei an das Ende der bestehenden Dateienliste angefügt.

act Im Action-Feld der am Ende der Dateienliste hinzugefügten Select-Datei wird der angegebene Action-Code eingetragen.

Zum Thema "Selektieren von Bibliothekselementen" siehe auch Variable Action ONXSEL (Seite **Fehler! Textmarke nicht definiert.**, sowie Action-Code S (Seite **203**).

Einfügeposition für kopierte Dateien

SET COPYMODE=A|E Einfügeposition der durch den Action-Code C neu erzeugten Kopien in der Dateienliste:

A direkt hinter der Ursprungsdatei

E am Ende der Dateienliste.

Standard: SET COPYMODE=E.

Behandlung von Zeichenfolgen im Suche-Kommando

SET SEARCHMODE=A|C

A Im Suche-Kommando wird '...' bzw. C'...' wie A'...' behandelt.

C Im Suche-Kommando wird '...' bzw. C'...' nicht wie A'...' behandelt.

Standard: SET SEARCHMODE=C.

Verhalten beim Suchen mit Ersetzen von Zeichenfolgen

SET REPLACEMODE=L|F

L Bei einem Suche-Kommando der Art S,'123'*'890'='abc' wird bei Treffern die Zeichenfolge '890' durch 'abc' ersetzt.

F Bei einem Suche-Kommando der Art S,'123'*'890'='abc' wird bei Treffern die Zeichenfolge '123' durch 'abc' ersetzt.

Standard: SET REPLACEMODE=F.

SET-Kommando: CFS-Parameter festlegen

SET Es wird eine Maske ausgegeben, in der die Belegungen der am häufigsten benutzten Parameter angezeigt und für den aktuellen Lauf verändert werden können.

SET [%*name* | %? [,*jrsave*]]

Die Inhalte der in der SET-Maske aufgeführten Parameter werden mit den in dem Parameterset %*name* gespeicherten Angaben modifiziert. Die Ausgabe der SET-Maske wird unterdrückt.
%?: Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

Das Abspeichern der zuletzt verwendeten SET-Parameter in einen Parameterset erfolgt mit dem Kommando SP SET%*name*. Näheres hierzu auf Seite 283.

Die aufgrund des SET-Kommandos ausgegebene Maske hat folgendes Format:

```
                              C F S   Parameter-Settings

   User-Library                      (UL): CFS.USERLIB
   Standard User-Option(s)           (UO):
   EDT Initialisation-commands   (EDTINIT):
   Standard User-ID                 (UID):
   Standard Sort-Option              (SO): NAME,A     Hardcopy    (HC/NHC): OFF
   Query on  Copy: OFF   Add: OFF   Select: ON   Erase: OFF   Term: OFF  EDT: ON
   Show used resources  (M/CPU/WSI/IO/OFF): OFF  Display          (AGE/DATE): AGE
   Erase with Retain Tempfiles  (ERT/NERT): ON   Display       (LASTP/ALLOC): LASTP
   Duplicate Key mode         (DUPK/NDUPK): OFF  PLAM :  Age as  (SDAT/UDAT): SDAT
   Command-characters for Isam-Editing (RD)      Copy : C    Elim : E  Retain: R
              Insert: I   Move : M   Join : J    Clear: *    After: A  Before: B
   Transform lowercase letters   (LOW/CAP): CAP CFS Command-Separator  (SEP): ;
   Check PLAM-Types on Add    (CT/CTQ/NCT): CTQ INF Standard-Format (I FORM):

                                Connection: 0 1 2 3 4 5 6 7 8 9
   Direct K2-mode ON/OFF (1/0)  (DKCn/NDKC): 0 0 0 0 0 0 0 0 0 0
   Asynchron Mode                (ACn/NACn): 0 0 0 0 0 0 0 0 0 0
   Hardcopy in Connections       (HCn/NHCn): 0 0 0 0 0 0 0 0 0 0
   Key for single Mask HC in Conn.     (CH): NONE Show Conn. Line    (SCL/NSCL): OFF
   Automatic Connection Display (ACD/NACD): ON   DO-Params in Conn.    (DP/NDP): ON
   Symbol to invoke Command-Memory        : -    Escape-Symbol in  Connections: ^
   BS2000-Prompt in Connections        (CP): /
```

Hinweise:

Alle Parameter können auch mit den in der SET-Maske aufgeführten Einzelkommandos (xxx/yyy) gesetzt oder zurückgesetzt werden. Falls die Einzelkommandos in der SET-Maske nicht aufgeführt sind, so sind die Einzelkommandos gleichlautend mit den in Großbuchstaben angegebenen Alternativen, z.B. "Display AGE/DATE". In diesem Fall lauten die Einzelkommandos AGE/DATE. Die Einzelkommandos zum Umdefinieren der Zeichen für das Kommandogedächtnis und das Connection-Fluchtsymbol sind auf Seite 351 und 374 beschrieben.

Nicht für alle in CFS einstellbaren Parameter existiert in der SET-Maske ein entsprechendes Feld bzw. ein Einzelkommando. Diese sind im allgemeinen RZ-globale Parameter und können nur über das Modify-Kommando von CFS (M L'...',C'...') benutzerindividuell angepaßt werden. Mehr hierzu siehe Seite 479.

Das Setzen von benutzerspezifischen Parametern kann automatisiert erfolgen durch Eintragen der Einzelkommandos in eine CFS-Startup-Datei.

Beispiel für das Setzen von Parametern mit einer Startup-Datei:

`*002 FILE=NO`	Verzweigen zum Kommandofeld der zweiten CFS-Maske
`UO EXTNT`	Default User Option definieren
`ALLOC`	Anzeige von Allocated Space anstelle Last-Page
`SO AGE,A`	Sort Option: Dateienliste aufsteigend nach AGE sortieren
`DUM`	Display Used Memory anstelle der Uhrzeit
`CP '&CON /'`	Connection Prompt: n / (n = Conn.-Nr)
`CRD -=§`	Zeichen für Kommandogedächtnis in Connections ändern
`NP`	Verzweigen in die CFS-Selektionsmaske

Für eine ausführliche Darstellung der Möglichkeiten des Startup-Mechanismus von CFS wird auf Seite 537 und folgende (Startup) verwiesen.

Show Item: Text zu Termin anzeigen

SI [*datum*]

Show Item. Es werden alle gespeicherten Informationen zu dem gewünschten Termin angezeigt. Eine ausführliche Beschreibung des Kommandos finden Sie auf Seite 427 und folgende (Terminverwaltungssystem).

Show Item List: Terminliste

SIL [*datum*] [+|-] | [{+|-} *n*] | *such*

Show Item List. Es werden die Kurzinformationen zu allen Terminen in einem bestimmten Zeitbereich angezeigt. Es können auch alle Termine gesucht werden, die einen vorgegebenen Suchbegriff in ihrem Beschreibungstext enthalten. In der SIL-Maske können mit den angezeigten Terminen verschiedene Aktionen ausgeführt werden, z.B. Termin verschieben, löschen, Bearbeitungsvermerk setzen.

Eine ausführliche Beschreibung des Kommandos finden Sie auf Seite 427 und folgende (Terminverwaltungssystem).

Show Item Short: Kurzübersicht der Termine in der Systemzeile

SIS [*datum*] [+|-] | [{+|-} *n*]

Show Item Short. In der 25-ten Zeile (Systemzeile) wird das Datum zu allen Terminen im gewünschten Zeitbereich ausgegeben. Das SIS-Kommando eignet sich besonders zur Verwendung in Startup-Prozeduren von CFS. Eine ausführliche Beschreibung des Kommandos finden Sie auf Seite **427** und folgende (Terminverwaltungssystem).

Angezeigte Dateienliste sichern

SL [*mn*] Save List. Die aktuelle Dateienliste wird in einer Datei gesichert.

mn ist eine 1- bis 8-stellige Kurzkennung (Mnemo-Code). Durch RL *mn* - eingegeben als Kommando (siehe oben), bzw. im Feld 'FILENAME-SELECT' der Selektionsmaske - wird die gesicherte Dateienliste wieder angezeigt.

Der Name der Sicherungsdatei ergibt sich als #CFS.SAVELIST., wobei der im SL-Kommando angegebene Mnemo-Code als Suffix angehängt wird. Der feste Teil des Namens #CFS.SAVELIST. kann vom Systemverwalter abgeändert werden. Die mit SL gesicherten Dateienlisten sind temporäre Dateien und werden bei Prozeßende vom BS2000 gelöscht.

Falls die mit SL gesicherten Dateienlisten auch nach LOGOFF noch für eine spätere Bezugnahme bestehen bleiben sollen, so ist bei der Installation von CFS durch den Systemverwalter die Prefix #CFS.SAVELIST. durch CFS.SAVELIST. zu ersetzen.

Falls im SL-Kommando kein Mnemo-Code angegeben wurde, so wird der Jobname des Benutzerprozeßes und falls dieser nicht existiert, die TSN als Default Mnemo-Code verwendet. Das gleiche gilt für das Kommando RL (Restore List).

Das SL-Kommando hat die gleiche Funktion wie das Feld DOCUMENTATION in der Selektionsmaske von CFS. Im Feld DOCUMENTATION können jedoch noch einige zusätzliche Optionen angegeben werden.

Kommandogedächtnis sichern

SM [*datei* | [*bibl*] ([*t*/] *element*)]

Save Memory. Der Inhalt des CFS-internen Gedächtnisses (Eingaben in der Selektionsmaske von CFS, Kommandoeingaben, Eingaben in Connections) wird in einer Datei / einem PLAM-Bibliothekselement gesichert.

datei Name der Datei, in der das Kommandogedächtnis gesichert wird.

[*bibl*] ([*t*/] *element*)

Name der Bibliothek und des Elements, in dem das Kommandogedächtnis gesichert wird. Standard für die Bibliothek: CFS-Userlib. Standard für den Elementtyp: S (Source).

Wird im Kommando SM weder ein Datei-, noch ein Elementname angegeben, so erfolgt die Sicherung des Kommandogedächtnisses in eine Datei mit dem Namen CFS. MEM.STD.*jobname* bzw. CFS.MEM.STD.*acct*, falls beim LOGON kein Jobname angegeben wurde.

Format der Save Memory-Datei:

Das Kommandogedächtnis besteht aus zwei bis 12 verschiedenen Abschnitten:
1) Eingaben in der Selektionsmaske von CFS. Die Eingaben zu den einzelnen Feldern werden durch das Zeichen ` (Tabulator) getrennt.
2) Eingaben im Kommandofeld der Dateienliste.
3) Eingaben in Connection 0.
   ....
12) Eingaben in Connection 9.

Jeder Abschnitt des Kommandogedächtnisses wird in der SM-Datei durch einen Header-Satz der Art: "$FILENAME-SELECT:", "$COMMAND:", "$CON0:", ..., "$CON9:" eingeleitet. Jeder Eintrag im Kommandogedächtnis wird sodann in einem eigenen Datensatz in der SM-Datei abgelegt.

Für weitere Informationen zum Thema Kommandogedächtnis siehe Seite **211**.

Standard Sort Option für Dateienliste festlegen

SO [(*c*,*l*)] | AGE | <u>NAME</u> | NAME(*c*,*l*) | SIZE | USER | NONE] [, <u>A</u> | D]

Die Dateienliste wird standardmäßig nach Namen aufsteigend sortiert angezeigt (Standard Sortierkriterium). Mit dem Kommando SO hat der Benutzer die Möglichkeit, ein anderes Feld als den Dateinamen als primäres Sortierkriterium, sowie die Reihenfolge der Sortierung (auf-/absteigend) festzulegen. Felder mit dem gleichen primären Sortierkriterium werden in zweiter Instanz automatisch nach Namen aufsteigend sortiert. Mit SO wird das im Feld SORT OPTION der Selektionsmaske angezeigte Standard Sortierkriterium vorbelegt. Eine Eingabe im Feld SORT OPTION (siehe Seite **72**) wirkt nur temporär und überschreibt nicht das durch SO festgelegte Standard Sortierkriterium.

Für eine ausführliche Beschreibung der Parameter des Kommandos SO siehe Seite **282**.

SO? Es wird die aktuell gültige Sortieroption angezeigt.

Dateienliste umsortieren

SORT [*sort-opt*]

Die aktuelle Dateienliste wird neu sortiert. Es findet hierbei keine neue Selektion statt. Falls keine Sort-Option angegeben ist, wird bezüglich des zuletzt verwendeten Kriteriums sortiert. Für *sort-opt* können die gleichen Optionen angegeben werden wie im Feld SORT-OPTION der Selektionsmaske (siehe Seite **72**) oder im Kommando SO (siehe Seite **482**).

sort-opt (*c*,*l*) | AGE | NAME [(*c*,*l*)] | SIZE | USER | NONE [, <u>A</u> | D]

(*c*,*l*) Sortierung nach einer absoluten Spaltenangabe (*c* = Column, *l* = Länge).
Die Spaltenangabe bezieht sich auf die Spalten eines in der Dateienliste eingeblendeten Zeilenlineals (Kommando SC).
Durch die Angabe (8,1) kann z.B. nach der Typ-Angabe sortiert werden.

AGE Die Dateienliste wird gemäß dem Inhalt in der AGE-, bzw. DATE-Spalte sortiert. Bei PLAM-Bibliotheken wird die Uhrzeit der letzen Änderung (TIME) im Rahmen der Sortierung mit berücksichtigt.

NAME [(*c*,*l*)] Die Dateienliste wird bezüglich der Namen der Datenobjekte sortiert (Standard). Über den Zusatz (*c*,*l*) kann ein Teil des Namens als Sortierfeld festgelegt werden. *c* gibt dabei die Spalte innerhalb des Namens und *l* die Länge des Namensteils an.
Standard: NAME(1,44).

Hinweis:

Bei PLAM-Bibliotheken kann durch die Sort Option NAME(1,44) eine Sortierung erreicht werden, bei der Elemente mit gleichen Namen und verschiedenen Typ-Bezeichnungen aufeinander folgen.
Durch die Sort Option NAME ohne Zusatz wird bei PLAM-Elementen der Elementtyp bei der Sortierung der Elementnamen mit berücksichtigt. Es ergibt sich in diesem Fall ein anderes Sortierergebnis als bei der Angabe NAME(1,44).

SIZE Die Dateienliste wird gemäß dem Inhalt der ersten Spalte (LASTP, bzw. ALLOC bei Dateien und SPACE bei Bibliotheken) sortiert.
Anstelle von SIZE wird auch LASTP, ALLOC und SPACE akzeptiert.

USER Falls bei der Selektion der Dateienliste eine USER OPTION angegeben wurde, so wird durch die Sortieroption USER eine Sortierung gemäß dem in der User Option-Spalte enthaltenen Wert vorgenommen. Falls keine USER OPTION angegeben wurde, so bleibt diese Sortieroption ohne Wirkung.

NONE Die Dateien-/Elementliste wird unsortiert ausgegeben.

A Aufsteigende Sortierreihenfolge (Standard).

D Absteigende Sortierreihenfolge.

Hinweis

Bei expliziter Angabe einer Sort-Option wird die Sortierung auf die gesamte Dateienliste angewendet, auch wenn diese schrittweise durch mehrere AL-Kommandos erzeugt wurde. Im Standardfall ist der zu sortierende Bereich auf jeweils ein NP-/ AL-Kommando beschränkt.

Beispiele:

`SORT AGE` oder `SORT DATE`
Die bestehende Dateien- bzw. Elementliste wird nach dem Alter der Datenobjekte aufsteigend sortiert.

`SORT SIZE,D`
Die aktuelle Dateienliste wird bezüglich der LASTP-Angabe absteigend sortiert.

Parameterset speichern

SP [COMP | FT | GUARD | SET | VAR] %*name* [,*jrfile*]

Es wird ein Parameterset gespeichert. Der Inhalt des Parametersets ist die zuletzt angegebene Selektionsbedingung, die zuletzt verwendeten Parameter des File-Transfer, Compare bzw. SET-Kommandos.

COMP | FT | GUARD | SET | VAR

Es werden die zuletzt verwendeten COMP-/FT-/GUARD-/SET-Parameter gespeichert. Bei FT werden auch evtl. definierte FTEP-Parameter berücksichtigt.
Bei VAR wird die zuvor definierte Variable Action gesichert. Bei ONX/*cmd-name* ? wird auch der Inhalt des SDF-Fragebogens gesichert. Die Variable Action kann spä-

ter aus dem gespeicherten Parameter Set wieder aktiviert werden durch ONX%*name* [,*jrfile*]
Standard: Es werden die zuletzt verwendeten Selektionsbedingungen von CFS gespeichert.

%name Name des Parametersets, in den die Parameter gespeichert werden sollen.

jrfile Name der JRSAVE-Datei, in der der Parameterset gespeichert wird.
Standard: CFS.JRSAVE. ... Näheres siehe Seite 565.

Passwort für CFS-Benutzungssperre vorgeben

SPL [*passw*] Set Password for Lock. Es wird ein Passwort für ein später folgendes LOCK-Kommando definiert. Wird *passw* im SPL-Kommando nicht angegeben, so kann der Benutzer das Passwort in einem dunkel gesteuerten Feld eintragen.

Status aller Job-Report Aufträge anzeigen

STAJ [*jr-id*] Status Job Report.

Status der mit dem E-Kommando gestarteten Enter-Aufträge anzeigen.

Für eine ausführliche Beschreibung des Kommandos STAJ, des E-Kommandos und aller weiteren Job-Report Kommandos (JR/ JRID/ JRF/ JRIC/ JRP/ JREP/ JRSH) wird auf Seite 397 und folgende (CFS Job-Report System) verwiesen.

Bildschirmgröße für Windows-Terminalemulation festlegen

SWS BS2000-Terminalemulationen unter Windows sind nicht an die 24*80 Beschränkung von 9750-Terminals gebunden. Mit diesem Kommando wird die Bildschirmgröße des Emulationsfensters, insbesondere die Anzahl der darstellbaren Zeilen festgelegt. Eine größere Zeilenanzahl kann z.B. bei der Anzeige der Dateienliste in CFS und im Display-Modus optimal angewendet werden.

Es stehen zwei verschiedene Betriebsarten des Kommandos SWS zur Verfügung:

1) SWS für beliebige TCP-IP Emulationen im **9763-Modus**:

SWS2 Analog zum Kommando VF2 des EDT wird eine evtl. angezeigte Display-Datei (Action-Code D/M) in einem Fenster mit 27 Zeilen und 132 Spalten angezeigt. Auch die Protokollausgabe der Job-Report Maske erfolgt in einem breiten Fenster.
Falls der unicodefähige EDT V17 eingesetzt ist, wird bei dem Action-Code EDT und dem Kommando EDT automatisch die Zeilen- und Spaltenanzahl im EDT mit der Anweisung VF2 auf 27, 132 eingestellt. Die Dateienliste von CFS wird nach wie vor mit 24 Zeilen und 80 Spalten dargestellt.

SWS3[W] Analog zum Kommando VF3 des EDT wird die Dateienliste von CFS in einem Fenster mit 32 Zeilen angezeigt. Auch alle Listen in TAS, JES und PUBSP werden in der entsprechenden Größe dargestellt. Falls der unicodefähige EDT V17 eingesetzt ist, wird bei dem Action-Code EDT und dem Kommando EDT automatisch die Zeilenanzahl im EDT mit der Anweisung VF3 auf 32 eingestellt.

W Dieser Zusatz bewirkt, daß der breite Display-Modus (132 Spalten) auch bei einem 32-zeiligen Bildschirm eingeschaltet bleibt. Dies ist keine garantierte Funktion und bewirkt bei manchen Emulationen (z.B. EM9750) einen Bildschirmüberlauf.

SWS4[W] wie SWS3, nur werden hier analog zum VF4-Kommando des EDT 43 Zeilen pro Bildschirm für die Dateienliste von CFS eingestellt.

2) SWS für TCP-IP Emulationen von com2, intrasys, ccom (beliebige Terminaltypen, insbesondere 9750): Hier kann die Zeilenanzahl des Bildschirms stufenlos eingestellt werden.

SWS *nn* Eine Voraussetzung zur Nutzung dieses Modus ist die Verwendung einer Windows-Terminalemulation unter TCP-IP, die das dafür notwendige erweiterte Protokoll versteht. Derzeit ist dies mit der TCP-IP Emu der Firma Com2, Intrasys und CCOM möglich.
Falls der SWS-Modus an Ihrer Emulation noch nicht verfügbar ist, wird eine entsprechende Fehlermeldung ausgegeben.

nn beliebige Zahl zwischen 16 und 60. Dieser Wert definiert die Anzahl der Zeilen, die von der Terminalemulation in einem Fenster ausgegeben werden.

NSWS Es wird auf den 24*80 Zeichenmodus (Standard) zurückgeschaltet.

Hinweise:

Bei SWS > 46 kann es in seltenen Fällen vorkommen, daß die Bildschirmmaske im unteren Teil abgeschnitten wird. In diesem Fall kann die Bildschirmmaske von CFS nicht auf eine Länge < 4096 Bytes komprimiert werden (CSO-Funktion: Compress Screen Output).

Falls SWS mit einem Wert größer 46 in einem CFS, das in einer Connection läuft aktiviert wird, so kann der Hardcopy-Modus (HC-Kommando) nicht genutzt werden.

PLAM-Bibliotheken: Age aus "User-Date" bestimmen

UDAT PLAM-Bibliothekselemente: Die Age-Angaben in der Elementliste werden aufgrund des User-Datums errechnet.
Standard: UDAT.

Hinweise:

Beim Reorganisieren von PLAM-Bibliotheken wird das System-Datum aller Elemente auf den aktuellen Tag gesetzt. Das User-Datum wird beim Reorganisieren nicht verändert.

Bei Aufnahme einer Datei als Element in eine PLAM-Bibliothek wird von CFS das User-Datum des Bibliothekselements auf das Change Date der Datei gesetzt. Dies kann zur Folge haben, daß im UDAT-Modus neu aufgenommene Dateien mit einem Age-Wert > 0 in der Bibliotheksliste von CFS erscheinen.

Userlib auf andere Bibliothek zuweisen

UL *bibliothek* Die angegebene Bibliothek wird fortan als CFS-Userlib verwendet. Eine Information über die aktuell zugewiesene Userlib und das Zuweisen einer anderen Bibliothek kann auch in der Maske des SET-Kommandos erfolgen.
Standard: CFS.USERLIB (eigene Benutzerkennung)

Für weitere Informationen zum Thema CFS-Userlib wird auf Seite 541 und folgende (Userlib) verwiesen.

Standard User Option für die Selektionsmaske definieren

UO [*u-opt-1*] [; [*u-opt-2*]] [; [*u-opt-3*]] [; [*u-opt-4*]]

Die angegebene User Option bzw. die angegebenen User Options werden fortan als Standardwert in der Selektionsmaske von CFS eingetragen. Eine Information über die aktuelle Standard User Option und das Zuweisen einer anderen Standard User Option kann auch in der Maske des SET-Kommandos erfolgen. Standardmäßig ist keine User Option definiert.

Hinweise:

User Options bewirken, daß zusätzlich zu den von CFS standardmäßig angezeigten Dateimerkmalen noch ein oder mehrere weitere, vom Benutzer frei wählbare Merkmale in der Dateienliste ausgegeben werden. Durch Angabe einer Auswahlbedingung können Datenobjekte bezüglich dieser Merkmale auch selektiert werden. Mehrere User Options sind durch Semikolons zu trennen. Es können maximal vier User Options angegeben werden.

Das Kommando UO ohne Operanden macht die zuletzt definierten Standard User Options ungültig.

Das Kommando UO darf nicht mit weiteren Kommandos durch Semikolon ";" verkettet werden.

Der Wert für die als erstes angegebene User Option wird in einer eigenen User Option Spalte in der Dateienliste eingetragen. Die Werte für eine evtl. angegebene zweite, dritte und vierte User Option überschreiben in der Dateienliste die Spalten PASS, AGE und LASTP (SPACE).

Zwei aufeinanderfolgende Semikolons ";;" bewirken, daß die entsprechende Spalte der Dateienliste nicht mit einer User Option überschrieben wird. Siehe hierzu das zuletzt angegebene Beispiel.

Eine mit UO festgelegte User Option wirkt erst bei der nächsten Selektion. Um die neue User Option für die aktuelle Liste wirksam werden zu lassen, kann das Kommando NP* angewendet werden. NP* zeigt die neue User Option jedoch dann nicht an, falls in der aktuellen Selektion bereits eine andere User Option definiert war.

Beispiele:

`ALLO2;ALCNT;EXTNT;ALLOC`
Hiermit wird die User Option Spalte mit der Größe der Sekundärzuweisung (ALLO2) gefüllt. Die Spalte PASS zeigt die Anzahl der Sekundärzuweisungen (ALCNT), die Spalte AGE die Anzahl der Extents (EXTNT) und die Spalte LASTP die Anzahl Seiten, die der Datei zugewiesen wurden (ALLOC).

`ALCNT;EXTNT;;ALLOC`
Hiermit wird die User Option Spalte mit der Anzahl der erfolgten Sekundärzuweisungen (ALCNT) und die Spalte PASS mit der Anzahl der Extents (EXTNT) gefüllt. Die Spalte AGE wird nicht überschrieben (;;). Die Spalte LASTP wird mit dem der Datei zugewiesenen Platz (ALLOC) gefüllt.

Standard User-ID für die Selektion definieren

UID [*:x:* | *:*:* | *user-id* | *:x: user-id* | *:*: user-id*]

Die angegebene User-Id wird fortan als Standardwert in der Selektionsmaske von CFS eingetragen. Eine Information über die aktuelle Standard User-Id und das Zuweisen einer anderen Standard User-Id kann auch in der Maske des SET-Kommandos erfolgen.

user-id Selektion von Dateien im Default-Pubset der angegebenen Benutzerkennung. Die Benutzerkennung ist ohne $ und nachgestellten Punkt anzugeben.

Standard: keine standardmäßige User-Id.

Hinweis:

Das Kommando UID ohne Operanden macht die zuletzt definierte Standard User-Id ungültig. Eine mit UID festgelegte User-Id wirkt erst bei der nächsten Selektion.

Feld VARIABLE ACTION in der Selektionsmaske anzeigen/besetzen

VAR Feld VARIABLE ACTION der Selektionsmaske anzeigen.
Der Benutzer hat hier zwei Zeilen am Bildschirm zur Verfügung, in die er die gewünschte Variable Action eintragen kann.

VAR ONX/& *var-act*

Gleiche Wirkung wie ONX/& *var-act* : Das Feld VARIABLE ACTION wird mit dem angegebenen Inhalt gefüllt.

Plattenspeicherbelegung einer Datei anzeigen

VSNSP *datei* [, A] [, FILE=*listfile*] [, E]

Für die Datei wird der auf den einzelnen Volumes belegte Speicherplatz angezeigt.

A All. Es wird die Speicherbelegung für jeden einzelnen Extent der Datei angezeigt.

FILE=*listfile* [, E]

Die Ausgabe wird nicht am Bildschirm angezeigt sondern in die angegebene Datei geschrieben.
Der Parameter E (Extend) bewirkt, daß die Ausgaben an eine bestehende Listfile angehängt werden.

CFS in den Wartezustand versetzen

WAIT CFS wird in den Wartezustand versetzt. Der Wartezustand kann jederzeit beendet werden durch Betätigung der K2- oder einer anderen Datenübertragungstaste.

Im Wartezustand ist CFS unterbrechbar durch asynchrone Ereignisse, wie z.B. die Beendigung eines durch das E-Kommando gestarteten Enter-Prozeßes (Job-Report) oder durch Nachrichten, die aus Nebenprozessen (Connections) eingehen und von CFS zwischengespeichert müssen.

Hinweise:

Das WAIT-Kommando ist nützlich, wenn im Hintergrund Nebenprozesse (Connections) laufen, die sporadisch Ausgaben produzieren. Wenn der Benutzer im Grund-

prozeß oder in einer Connection längere Zeit keine Eingabe tätigt, kann es vorkommen, daß die Connection verloren geht, da die Nachrichten von CFS nicht rechtzeitig abgeholt werden können. Um dieser Situation vorzubeugen, ist es angezeigt, vor einer längeren Arbeitsunterbrechung CFS in den Wartezustand zu versetzen: Kommando WAIT in der Grundebene von CFS bzw. ^WAIT in Connections.

Das Wait-Kommando ist auch nützlich, wenn eine oder mehrere Connections auf den asynchronen Modus geschaltet sind (Kommando AC *n*), wie z.B. Nebenkonsolen (OC*n* $CONSOLE). Während des Wartezustands werden die Nachrichten aller asynchronen Nebenprozesse zum Zeitpunkt ihres Eintreffens am Bildschirm angezeigt.

Unsichtbare Einträge in Dateienliste sichtbar machen

YANK Alle mit dem Action-Code "-" unsichtbar gemachten Einträge in der Dateienliste werden wieder angezeigt.

Der Action-Code E bewirkt bei erfolgreicher Ausführung ebenfalls, daß die gelöschten Datenobjekte in der aktualisierten Dateienliste als unsichtbar gekennzeichnet werden. Mit dem Kommando YANK könnte nun noch nachträglich festgestellt werden, welche Dateien in der aktuellen Liste im einzelnen gelöscht wurden. Gelöschte Datenobjekte können bei eingeschalteter ERT-Option (Erase with Retain of Tempfiles) auch wieder zurückgeholt werden. Näheres hierzu auf Seite 474.

Durch einen von CFS eingetragenen Action-Code "-" werden auch alle Dateien aus der Liste entfernt, die mit einer Variablen Action ON&... erfolgreich bearbeitet wurden, sofern der Prozeßschalter 14 gesetzt ist (/SETSW ON=14). Näheres hierzu auf Seite 109. Mit YANK könnten die erfolgreich bearbeiteten Dateien wieder in der Dateienliste angezeigt werden.

8. CFS-Display/Editor

```
   dd.mm.yy   hh:mm:ss    HOST: ........   TSN: ....             USER-ID:  ........
   COMMAND :
   LASTP TYP FILENAME = $TSOS                              AGE PASS LACC  ACTION
      30 *I  xxxxxxxxxxxx  . . . . . . . . . . . . . . .   143 *       0 :D
       1:zzzzzzzzzzzzzzzzzzz.......................................................
       2:zzzzzzzzzzzzzzzzzzzz......................................................
       3:zzzzzzzzzzzzzzzzzzzz......................................................
       4:zzzzzzzzzzzzzzzzzzzzzzzzzzz...............................................
       5:zzzzzzzzzzzzzzzzzzzz......................................................
       6:zzzzzzzzzzzzzzzzzzzzz.....................................................
       7:zzzzzzzzzzzzzzzzzzzzzz....................................................
       8:zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz............................
       9:zzzzzzzzzzzzzzzzzzzzzzzzz.................................................
      10:zzzzzzzzzzzzzzzzzz........................................................
      11:zzzzzzzzzzzzzzzzzzzzz.....................................................
      12:zzzzzzzzzzzzzzzzzzzzzzzzzzz...............................................
      13:zzzzzzzzzzzzzzzzzzzzzzzzzzzzz.............................................
      14:zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz.....................................
      15:zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz.................................
      16:zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz......................................
      17:zzzzzzzz..................................................................
      18:zzzzzzzzzzzzzz............................................................
      19:zzzzzzzzzzzzzzzzzzzzzzzzzz................................................
   Help:  ? / ?cmd / ?all / ??
```

Im Display-Modus können alle von CFS unterstützten Datenobjekte (Dateien/Jobvariablen/Bibliothekselemente) angeschaut und modifiziert werden. Der Display-Modus wird eingeschaltet, indem in der Dateienliste bei dem gewünschten Datenobjekt der Action-Code D (Display) bzw. M (Modify) eingetragen wird. Im Display-Modus können im Kommandofeld der CFS-Maske alle bisher dargestellten Kommandos eingegeben werden. Darüber hinaus sind noch eine Reihe zusätzlicher Kommandos, die sog. Display-Kommandos, möglich. Die Display-Kommandos werden in den folgenden Abschnitten näher beschrieben.

Nicht abdruckbare Zeichen werden im CFS-Display durch das "Schmierzeichen" X'07' dargestellt. Die Entscheidung, welche Zeichen als "abdruckbar" gelten, wird getroffen aufgrund des CCS der Datei sowie des Terminaltyps, auf dem CFS läuft.

Läuft CFS auf einem nicht 8-bit fähigen Terminal, werden grundsätzlich alle nicht ein einem 7-bit Code definierten Zeichen als Schmierzeichen dargestellt.

Im Display-Modus werden bei Dateien mit einem 8-bit CCS (z.B. EDF041) auf einer Terminalemulation vom Typ 9763-8 (8-bit) alle in diesem CCS definierten Zeichen korrekt angezeigt.

Gemäß der Standard Umsetztabelle werden Kleinbuchstaben auch dann am Bildschirm dargestellt, wenn der LOW-Modus nicht eingeschaltet ist (siehe hierzu die Kommandos CAP/LOW).

Unterstützung von Unicode Objekten in CFS

Ab der OSD-Version 7.0 werden Unicode-Objekte unterstützt.
Unter Unicode-Objekten verstehen wir Dateien/PLAM-Bibliothekselemente, deren Datensätze in einem der folgenden CCS (Coded Character Sets) codiert sind: UTF16 (= UNICODE), UTF8, UTFE. Die Codes der alphanumerischen Zeichen (A - Z, a -z, 0 - 9) sind in UTF16 und UTF8 grundsätzlich verschieden vom EBCDIC-Code. Unter UTFE besitzen zumindest die alphanumerischen Zeichen die gleichen Codes wie in EBCDIC.

Im Display-/Modify-Modus von CFS werden die Inhalte von Unicode-Objekten in lesbarer Form dargestellt. Voraussetzung hierfür ist OSD >= 7.0 (XHCS Version >= 2.0), sowie der zur Codierung passende CCS-Eintrag im Systemkatalog bzw. im Inhaltsverzeichnis der PLAM-Bibliothek. Siehe hierzu auch die CFS User Option CCS.

DA UNI,EDF0xxx

Erkennt CFS eine Unicode-Datei am zugehörigen CCS-Eintrag, so wird intern das Kommando DA UNI,EDF04F ausgeführt. Der Benutzer kann jedoch selbst ein anderes Kommando DA UNI,EDF04xx eingeben, falls z.B. das Euro-Zeichen nicht an der erwarteten hexadezimalen EBCDIC-Position steht.

Das Kommando DA UNI,EDF04xx bewirkt folgendes:

a) Läuft CFS unter einer 8-bit Emulation, so werden alle Unicode-Zeichen, die im 8-bit Character Set EDF04xx darstellbar sind, zur Bildschirmanzeige in diesen Code umgesetzt.
b) Läuft CFS unter einer 7-bit Emulation, so werden alle Unicode-Zeichen, die im 7-bit Character Set EDF03IRV darstellbar sind, zur Bildschirmanzeige in diesen Code umgesetzt.

Alle Unicode-Zeichen, die im gewählten Code nicht definiert sind, werden als Schmierzeichen (X'08') dargestellt.

Im Hexadezimal-Modus von CFS (Display-Kommando H) werden die Daten der Unicode-Datei in der EDFxxx-Codierung angezeigt. Damit können auch im Hexa-Modus keine Unicode-Zeichen eingegeben werden, die außerhalb der EBCDIC Codevarianten liegen.

Um ein Unicode-Objekt im native-Modus, d.h. in der originalen, nicht umgesetzten Codierung anzuschauen, ist das CFS-Kommando NDA einzugeben. Bei UTF16-Dateien wird in diesem Fall jedes Zeichen als 2-Byte Folge dargestellt. Die gewohnten alphanumerischen Zeichen sind dann für den Benutzer nicht mehr erkennbar. Im Hexadezimalmodus können jedoch, falls bekannt, die Codierungen von fremdsprachlichen Zeichen (z.B. Kyrillisch) modifiziert werden.

DA CODE,UTFxxTOEBC

Mit diesem Kommando können die alphanumerischen Zeichen von Unicode codierten Dateien auch ohne korrekten CCS-Eintrag im Katalog, bzw. in älteren OSD-Versionen < 7.0 in einer lesbaren Form am Bildschirm angezeigt werden. Eine Änderung der Dateninhalte ist hierbei jedoch nicht möglich.

Falls eine Datei keinen, bzw. einen falschen CCS-Eintrag im Katalog besitzt, kann man mit

DA CODE,UTF16TOEBCD
DA CODE UTF8TOEBCD

den tatsächlich für die Codierung der Daten verwendeten CCS testen und der Datei evtl. mit /MOD-FILE-ATTR *datei*,C-C-S=*code* zugewiesen.

Es ist zu beachten, daß bei der UTF16 Codierung alphanumerische Zeichen als 2-Byte mit führendem Leerzeichen dargestellt werden.

Inhalt von Dateien/Bibliothekselementen umcodieren

ONX**CONV** ['*str1*'= '*str2*',] CCS=*ccsneu* [, FCCS=*ccsalt*]

Die Sätze der durch den Action-Code X gekennzeichneten Dateien bzw. PLAM-Bibliothekselemente werden gemäß dem angegebenen Coded Character Set Namen umcodiert.

CCS=*ccsneu*

im System gültiger Coded Character Set Name.
Der Inhalt der Datenobjekte wird mittels XHCS auf den angegebenen CCS umcodiert.
Falls eine Datei im Katalog bereits mit einen CCS *ccsx* verknüpft ist, wird angenommen, daß die Sätze in diesem CCS vorliegen und es erfolgt eine Umcodierung in der Form *ccsx* --> *ccsneu*

FCCS=*ccsalt*

Falls die Datei im Katalog mit keinem CCS verknüpft ist, wird mit FCCS angegeben, in welcher Codierung die Eingabesätze vorliegen. In diesem Fall erfolgt die Umcodierung in der Form *ccsalt* --> *ccsneu*

Dateiinhalte suchen und Treffersätze wegschreiben

ONX**FIND** '*str*' [= W *datei*]

Die Suchzeichenfolge *str* wird unabhängig von der Codierung auch in UTFxx Dateien gefunden. Dies wird erreicht, indem die UTF codierten Datensätze vor der Suche ohne Datenverlust in den EBCDIC-kompatiblen Zeichensatz UTFE umgesetzt werden.
Treffersätze werden in diesem Code in die Trefferdatei geschrieben. Dies hat zur Folge, daß die alphanumerischen Inhalte im EBCDIC-Umfeld gelesen und ohne Verlust geändert werden können.

Rewrite-Kommando für geänderte Trefferdateien

REWR *datei* Beim Zurückschreiben von Treffersätzen in die Originaldateien/Bibliothekselemente wird zuerst geprüft, ob das originale Datenobjekt in einem Unicode-Zeichensatz (UTFxx) vorliegt. Ist dies der Fall, so werden die entsprechenden Treffersätze von UTFE nach UTFxxx umgewandelt und erst danach in das Datenobjekt zurückgeschrieben.

Action-Code EDT / Kommando EDT

Falls unter OSD 7.0 der neue EDT V17.0 eingesetzt ist, kann dieser von CFS mit der erweiterten Funktionalität benutzt werden. Diese beinhaltet:

a) Dateien mit Satzlängen > 256 Bytes werden automatisch ohne Datenverlust eingelesen und können mit UPD bzw. Write-Kommando zurückgeschrieben werden.

b) beliebige Dateien im Unicode Format können bearbeitet werden.

Standardmäßig verwendet CFS auch unter OSD 7.0 den EDT im kompatiblen 16.6-Modus mit allen gegebenen Einschränkungen. Es existiert jedoch ein Kommando zum Einstellen des Verarbeitungsmodus des EDT:

EDTMODE COMP | UNI

Vor dem ersten Aufruf des Kommandos / Action-Codes EDT wird festgelegt, ob dieser im kompatiblen Format oder mit der erweiterten Unicode-Funktionalität aufgerufen wird. Der durch das Kommando- bzw. Action-Code benutzte EDT kann somit in unterschiedlichen Modi betrieben werden. Nachdem EDT geladen wurde, ist eine erneute Änderung des Modus für die jeweilige Aufrufart (Kommando/Action-Code) nicht mehr möglich.

Der EDT-Aufrufmodus kann auch im Modul CFSMAIN global für alle Benutzer eingestellt werden. Das Kommando EDTMODE übersteuert diese globale Voreinstellung.

Kommandos zum Modifizieren von Datenobjekten

Erweiterte Editierfunktionen für sequentielle Dateien/Bibliotheks-elemente

IS [*:x:* | *vsn/device* | *step*]

Die am Bildschirm angezeigte SAM-Datei/sequentielles Bibliothekselement wird mit ISAM-Schlüsseln versehen. Dies geschieht durch Umsetzen der ursprünglichen Datei in eine ISAM-Zwischendatei.
Im ISAM-Format können neben der 1 zu 1 Änderung erweiterte Editierfunktionen angewendet werden, wie Kopieren, Einfügen und Löschen von Datensätzen, Einfügen und Löschen von Spaltenbereichen.

Die optionale Angabe eines PVS (*:x:*) oder einer Privatplatte (*vsn/device*) bewirkt, daß die Zwischendatei im angegebenen PVS bzw. auf der Privatplatte angelegt wird. Der Parameter *:x:* bzw. *vsn/device* wird intern gespeichert und bei allen folgenden Umsetzungen verwendet, sofern keine anderen Angaben gemacht wurden.

step Schrittweite für die Generierung der ISAM-Schlüssel. Standardmäßig werden die in der Zwischendatei eingefügten 8 Byte langen Schlüssel mit einer Schrittweite von 100 erhöht. Um sequentielle Dateien mit mehr als 1.000.000 Datensätzen in eine ISAM-Zwischendatei umzuwandeln, sollte im IS-Kommando eine Schrittweite von 10 angegeben werden.
Die Schrittweite wird intern gespeichert und bei allen folgenden Umsetzungen verwendet, sofern keine anderen Angaben gemacht wurden.

SAM Rückumsetzen der mit dem Kommando IS bearbeiteten ISAM-Zwischendatei in das sequentielle Datenformat.

Das Verlassen des Display-Modus durch Betätigen der K1-/F1-Taste hat die gleiche Wirkung wie das Kommando SAM. Der Benutzer wird gefragt, ob die bearbeitete ISAM-Datei in das SAM-Format zurückübertragen werden soll. Wünscht der Benutzer, daß die ISAM-Datei nicht in das SAM-Format überführt wird, so bleibt die SAM-Datei unverändert und die ISAM-Datei wird nicht gelöscht.

Kleinbuchstaben im Modify-Modus / Suche-Kommando

LOW | <u>CAP</u> Kleinbuchstaben / Großbuchstaben.

LOW CFS-Editor: Beim Modifizieren wird keine Umwandlung der eingegebenen Kleinbuchstaben in Großbuchstaben vorgenommen.

Eingabe von Dateidokumentationen (Action-Code IM): Der Text wird in der eingegebenen Groß-/Kleinschreibung übernommen.

Suche-Kommando, User Option FIND, Variable Action [ONX]FIND: Es erfolgt keine Umwandlung von Kleinbuchstaben im Suchargument in Großbuchstaben. Diese Funktion kann auch durch Eingabe des Suchарguments in der Form L'....' erreicht werden.

Action-Code EDT: Beim Einlesen einer neuen Datei/eines neuen Bibliothekselements wird der LOWER-Modus des EDT unverändert gelassen. Ein bei der Bearbeitung der letzten Datei eingegebenes LOWER ON-Kommando bleibt damit wirksam.

Job-Report Do-Parameter Maske: In Hochkommas eingegebene Parameter werden bezüglich der Groß- Kleinschreibung unverändert an die Prozedur übergeben und nicht, wie dies standardmäßig der Fall ist (CAP), in Großbuchstaben umgewandelt.

CAP | **NLOW** CFS-Editor: Beim Modifizieren werden Kleinbuchstaben vor dem Zurückschreiben in die Datei in die entsprechenden Großbuchstaben umgewandelt.

Eingabe von Dateidokumentationen (Action-Code IM): Der eingegebene Text wird in Großbuchstaben umgesetzt.

Suche-Kommando, User Option FIND, Variable Action ONXFIND: Suchargumente ([C]'*item*') werden in Großbuchstaben umgesetzt.

Action-Code EDT: Beim Einlesen einer neuen Datei/eines neuen Bibliothekselements wird der zuvor evtl eingeschaltete LOWER ON-Modus des EDT zurückgesetzt (LOWER OFF).

Job-Report Do-Parameter Maske: In Hochkommas eingegebene Parameter werden in Großbuchstaben umgewandelt.

Anstelle von CAP kann auch das Kommando NLOW oder LOWO (LOW Off) angegeben werden.

Standard: CAP.

Hinweis: LOW bzw. CAP dürfen nicht verkettet mit weiteren Kommandos angegeben werden. Die Kommandofolge LOW;M ist beispielsweise nicht erlaubt.

Datenobjekt zum Ändern freigeben

M | NM Modify/No Modify.

Mit dem Modify-Kommando wird der Inhalt eines mit dem Action-Code D (Display) angekreuzten Datenobjekts (Datei/Jobvariable/Bibliothekselement) zur Änderung freigegeben. Beim Markieren mit dem Action-Code M (Modify) ist das Kommando M automatisch impliziert. Die Modifizierung erfolgt bei Dateien und Jobvariablen sofort nach Betätigen der Datenübertragungstaste. Bei Bibliothekselementen werden die Änderungen bei Verlassen des Modify-Modus zurückgeschrieben.
Der Modify-Modus wird ausgeschaltet durch das Kommando NM (No Modify).

Standardmäßig können sequentielle Dateien und PAM-Dateien im CFS-Editor nur 1 zu 1 modifiziert werden. Für ISAM-Dateien stehen erweiterte Editierfunktionen zur Verfügung. Diese sind im Abschnitt "Editieren von ISAM-Dateien beliebigen Formats" beschrieben. Auch sequentielle Dateien können im ISAM-Format mit diesen erweiterten Funktionen editiert werden. Mehr hierzu auf Seite 335.

Hinweise:

Beim Modifizieren von Daten mit CFS darf die Taste "Ausfügen Zeichen" nicht benutzt werden, da sich hierdurch der gesamte am Bildschirm angezeigte Dateninhalt verschiebt.

Beim Modifizieren im Hexadezimalmodus können Zeichen auch in Character Darstellung eingegeben werden. Hierbei ist das gewünschte Zeichen im ersten Halbbyte einzugeben. An der Stelle des zweiten Halbbytes ist ein Leerzeichen anzugeben. Beispiel: Zur Eingabe des Zeichens 'F' (X'F6') kann der ursprünglich an dieser Stelle stehende Hexadezimalwert CA durch F␣ überschrieben werden. Es ist damit

auch möglich, Kleinbuchstaben im Hexa-Modus einzugeben. Dazu muß jedoch der LOW-Modus eingeschaltet sein.

Es ist zu beachten, daß im Unterschied zum EDT, Kleinbuchstaben im CAP-Modus von CFS (siehe oben) als solche dargestellt werden; eingegebene **Kleinbuchstaben** werden im Standardmodus CAP in Großbuchstaben umgesetzt.

An einem Datensichtgerät mit deutscher Tastatur werden im Gegensatz zu früheren CFS-Versionen folgende Zeichenumwandlungen im CAP-Modus <u>nicht</u> durchgeführt:

'{' ('ä') --> '[' ('Ä')
'|' ('ö') --> '\' ('Ö')
'}' ('ü') --> ']' ('Ü')

Zum Verkürzen von Jobvariablen muß zuerst der LOW-Modus (siehe oben) eingestellt werden. Anschließend sind die zu löschenden Daten der Jobvariablen (rechtes Ende der relevanten Daten) mit NIL-Zeichen zu ersetzen. D.h. es ist die AFG-Taste (Ausfügen) solange zu betätigen, bis der zu löschende Text und alle Zeichen rechts davon durch NIL-Punkte ersetzt sind.

Beim Modifizieren von ISAM-Dateien werden die Stellen vom Ende des Datensatzes bis zum Ende der Bildschirmzeile mit überschreibbaren Nil-Zeichen gefüllt. Die Datensätze können somit durch Überschreiben der Nil-Zeichen über das bisherige Satzende hinaus erweitert werden.
Das Verkürzen eines Datensatzes durch Überschreiben mit NIL-Zeichen ist nicht möglich.

Kommandos zum Anzeigen von Datenobjekten

Verschieben des Sichtfensters horizontal/vertikal

+ | -
Sichtfenster um einen Bildschirm weiter zum Ende/Anfang verschieben.

Neben "+" führt auch das leere Kommandofeld (abgesendet mit ENTER) auf den nächsten Bildschirm.

++ | --
Sichtfenster auf Ende/Anfang der Datei positionieren.

Hinweise:

Das Positionieren an das Ende einer **sehr großen SAM-Datei** kann wesentlich beschleunigt werden. Anstelle von ++ ist hierzu das Kommando PP*nnnn* (Position to Pam-Block, siehe Seite **306**) anzugeben. *nnnn* ist hier die Nummer des letzten Datenblocks (LASTP-Angabe). Die Satznummerierung geht in diesem Falle verloren.

Während der normalen Positionierung an das Ende einer sehr großen SAM-Datei (++) kann es zu längeren Wartezeiten kommen, da die Datei von CFS vollständig gelesen werden muß. Ein Abbrechen dieses Vorgangs wird erreicht durch Betätigen der K2-Taste (Verzweigen in den Kommando-Modus des BS2000) und anschließendes /INTR R Kommando. Das Sichtfenster wird in diesem Fall auf die zuletzt gelesene Stelle in der Datei positioniert.

Bei PAM-Dateien kann mit ++ nur bis zum letzten beschriebenen Datenblock positioniert werden. Unter TSOS kann im Display-Modus auch auf die PAM-Seiten jenseits des Last-Page Pointers positioniert werden.

+*n* | -*n*
Sichtfenster um *n* Sätze weiter zum Ende/Anfang verschieben.

R | L
Rechts/Links. Sichtfenster um eine Bildschirmbreite nach Rechts/Links verschieben (im NDL-Modus). Im DL-Modus wird das Sichtfenster um das Fassungsvermögen eines ganzen Bildschirms nach Rechts/Links verschoben.

> | <
gleiche Wirkung wie R/L.

RR | LL
Sichtfenster zum Satzende/Satzanfang positionieren.
Bei Sätzen unterschiedlicher Länge ist jeweils der erste, am Bildschirm angezeigte Satz maßgebend.

>> | <<
gleiche Wirkung wie RR/LL.

R *n* | L *n*
Sichtfenster um *n* Stellen nach Rechts/Links verschieben.

>*n* | <*n*
gleiche Wirkung wie R *n*/L *n*.

R X'*hex*' | L X'*hex*' | >X'*hex*' | <X'*hex*'
Sichtfenster um X'*hex*' Stellen nach Rechts / Links verschieben.

C *n*
Sichtfenster auf Spalte *n* positionieren. Satzanfang = C1.

CX'*hex*'
Sichtfenster auf Spalte X'*hex*' positionieren. Satzanfang = CX'0'.

Hinweis:

Als Alternative zur horizontalen Verschiebung des Bildschirmfensters sei auf das Kommando AD (Arrange Data) hingewiesen. Dieses Kommando ermöglicht die selektive Anzeige bestimmter Spaltenbereiche einer Datei. Die Reihenfolge der Spalten

und deren Länge ist frei bestimmbar. Für jeden ausgewählten Spaltenbereich kann ein Darstellungsformat (Character/Hexadezimal) gewählt werden.

Spaltenbereiche für die Anzeige auswählen/umorganisieren

AD { [*col* | *konst* | *seq*] ["*text*"] } {, ... } ... [, FC= X'*cc*' | C'*c*']

Arrange Data. Der Anzeigemodus der Display-Datei wird dahingehend verändert, daß nur bestimmte Spaltenbereiche der Datensätze ausgewählt und in einer festgelegten Reihenfolge am Bildschirm dargestellt werden. Zwischen den Spaltenbereichen können konstante Strings eingefügt werden. Die Darstellung jeder einzelnen Spalte kann in character und hexadezimaler Form erfolgen. Zu den ausgewählten Spaltenbereichen können Beschreibungstexte für die Überschriftszeile (Scale) definiert werden. Durch das Kommando AD wird ausschließlich die Darstellung der in der Display-Datei gespeicherten Datensätze beeinflußt. Die Datei selbst bleibt unverändert bestehen.
Ein evtl. aktiver Offset-Modus für die Datei (z.B. O-4) wird durch das AD-Kommando zurückgesetzt.

col Angabe des auszuwählenden Spaltenbereichs. Es sind zwei verschiedene Darstellungsweisen möglich.

:col1: len [B[S]|C|H|P[*n*]|Z|STCK]
Anfangsspalte des auszuwählenden Bereichs. Die Spaltenzählung beginnt mit 1.
Es kann auch eine negative Spaltenangabe gemacht werden (*:-col1:*). Damit kann z.B. das Längenfeld eines Satzes mit variabler Länge angezeigt werden: :-4:2B bzw. :-4:4X

len
Länge des Spaltenbereichs.

:col1-col2: [B[S]|C|H|P[*n*]|Z|STCK]
Definition des Spaltenbereichs durch die Anfangs- und Endespalte. Als Endespalte (col2) kann auch das Zeichen $ (= Satzende) angegeben werden.
Beispiel: :1-20: Es wird der Bereich von Spalte 1 bis Spalte 20 ausgewählt. Äquivalent hierzu ist die Angabe der Anfangsspalte :1: und der Länge 20 (:1:20).

B[S] Im ausgewählten Spaltenbereich steht eine Binärzahl mit (BS) oder ohne Vorzeichen (B). Die Länge des Spaltenbereichs darf nur 1, 2, 3 oder 4 betragen.

C Ausgabe des Spaltenbereichs im Characterformat (Standard).

H Ausgabe des Spaltenbereichs im hexadezimalen Format. H bewirkt einen auf die Spalte bezogenen Hexadezimalmodus (siehe Display-Kommando H/NH). Anstelle von H kann auch X angegeben werden.

P[*n*] Im ausgewählten Spaltenbereich steht eine gepackte Zahl mit oder ohne Vorzeichen.

n Anzahl der Kommastellen. Standard: *n* = 0
Beispiel: :35-38:P2 liefert z.B. folgendes Zahlenformat: 99999,50

Z Im ausgewählten Spaltenbereich steht eine Dezimalzahl mit Vorzeichen. Eine Zahl des Formats Z entspricht in COBOL der Definition "PIC S9" und ist hexadezimal in folgender Form gespeichert: F*a*F*b* ... F*n**Z**m*. Hierbei stehen *a*, *b*, ... ,*n*, *m* für Ziffern 0, ..., 9. *Z* enthält das Vorzeichen der Zahl: Die Sedezimalzeichen A, C, E, F stehen für positive und B, D für negative Werte.
Beispiel: F1C2 entspricht +12, F1D2 entspricht -12

STCK Im ausgewählten Spaltenbereich steht eine binär im STCK (Store Clock) Format gespeicherte Datums- und Zeitangabe. Das STCK-Feld kann vier oder acht Bytes lang sein. In der Ausgabe wird das Feld in der Form YYYY-MM-DD HH:MM:SS:MMMMMM angezeigt. MMMMMM entspricht hierbei dem Milli- und Mikrosekundenwert der Zeitangabe.

Standard: Die Spalte wird im Characterformat angezeigt.

konst Konstanter String, der in der aufbereiteten Satzdarstellung vor der nächsten Spalte, bzw. nach der zuletzt ausgewählten Spalte eingefügt wird.

[*len*] C'*string*' | X'*string*'

len
Gesamtlänge der Konstanten. Der in Form von C'...'/X'...' angegebene String wird solange wiederholt, bis die gewünschte Länge erreicht ist.
Standard: *len* = Länge des angegebenen Strings.

C'*string*' | X'*string*'
Wert der einzufügenden Konstanten. C'*string*' kann zu '*string*' abgekürzt werden.

seq Sequentielle Nummerierung. Bei SAM- und PAM-Dateien wird ein fortlaufendes Numerierungsfeld in die Anzeige eingefügt.
Bei ISAM-Dateien ist eine korrekte Ausgabe der Nummerierung nur garantiert, falls vom Anfang der Datei aus ein Write-Kommando oder ein Suche-kommando mit Wegschreiben der Treffer ausgeführt wird.
Falls mehrere SEQ-Einfügungen in einem Datensatz vorgenommen werden, muß für alle SEQ-Anweisungen die gleiche Länge und Stepsize angegeben werden.

SEQ(*len* [, *step*])

len
Länge des Numerierungsfeldes. 1 <= *len* <= 10.

step
Schrittweite der Numerierung. 1 <= *step* <= 30000.
Standard: 1

"text" Der angegebene Text definiert eine Spaltenüberschrift, die in der Scale-Zeile des Display-Bildschirms angezeigt wird.

Es können beliebig viele Spalten-/Konstantenbereiche aneinandergereiht werden. Die einzelnen Bereiche werden durch Kommas getrennt. Die ausgewählten Spalten /Konstanten werden in der Darstellung lückenlos aneinandergereiht. Mit dem Kommando AD können auch Spalten dupliziert oder vertauscht werden.

FC= Fill-Character. Spaltenbereiche, die ganz oder teilweise außerhalb eines Datensatzes liegen, werden mit diesem Zeichen aufgefüllt. Ausnahme: Der letzte Spaltenbereich wird nicht aufgefüllt, sondern entsprechend der Länge der vorhandenen Daten verkürzt. Für weitere Einzelheiten siehe Hinweise.
Standard: FC=C' '.

NAD AD-Modus ausschalten. Die Datensätze werden im Originalformat angezeigt.
Standard: NAD

AD Die mit dem letzten AD-Kommando definierte Spaltenauswahl wird wieder aktiviert.

AD? | AD?,%*name*

Die mit dem letzten AD-Kommando definierte Spaltenauswahl / eine mit dem Kommando SP %*name* in einem Parameter-Set gesicherte Spaltenauswahl wird angezeigt. Beschreibung von SP %*name* siehe unten.

AD ( )

Durch () wird eine spezielle Bildschirmmaske angefordert. In dieser stehen dem Benutzer 18 Zeilen zur Eingabe umfangreicher Spaltendefinitionen zur Verfügung. Diese sind lückenlos, d.h. ohne Blanks einzugeben.

Durch Betätigung der ENTER-Taste wird der AD-Modus aktiviert. Die K1-Taste bewirkt den Abbruch des AD-Kommandos. Mit der F3-Taste kann die ausgefüllte Maske in einer von CFS angelegten Hardcopy-Datei protokolliert werden. Ansonsten wirkt F3 wie ENTER.

Im Kommandofeld der Maske können BS2000-Kommandos sowie das SP-Kommando eingegeben werden.

SP %*name* [,*jrsave*]

Save Parameters. Mit diesem Kommando werden die in der AD-Maske (AD,()) eingetragenen Spaltenauswahl Definitionen unter einem Namen in der Datei CFS.JRSAVE [.*jobname*] bzw. in einer mit /FILE *datei*,LINK=JRSAVE zugewiesenen Datei gesichert. Die genaue Regel, nach der der Name der JRSAVE-Datei gebildet wird, ist auf Seite **564** unter CFS.JRSAVE beschrieben.

%*name* — ein- bis 44-stelliger alphanumerischer Name für den Parameter-Set.

jrsave — Name der JRSAVE-Datei. Der Dateiname sollte nur angegeben werden, wenn die Parameter in eine andere als die von CFS standardmäßig verwendete JRSAVE-Datei (siehe oben) gesichert werden.

Der Benutzer kann eine mit SP gesicherte Spaltenauswahl durch die Kommandos AD %*name* bzw. AD?,%*name* aktivieren/anzeigen lassen.

AD %*name* | %?

Es wird die Spaltendefinition aktiviert, die zu einem früheren Zeitpunkt mit dem Kommando SP %*name* in einem Parameter-Set gesichert wurde.
Die Inhalte der in der SET-Maske aufgeführten Parameter werden mit den in dem Parameterset %*name* gespeicherten Angaben modifiziert. Die Ausgabe der SET-Maske wird unterdrückt.
%?: Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

AD (*dat*) | (*bibl* ([*t*/] *elem*))

Die Spaltendefinitionen sind in einer sequentiellen Datei *dat* oder in einem Bibliothekselement gespeichert. Falls bei einem Bibliothekselement der Elementtyp nicht angegeben ist, wird S/ ergänzt, sofern es sich um eine PLAM-Bibliothek handelt.

Format der AD-Parameterdatei *dat*:
Jeder Datensatz beschreibt eine oder mehrere Spaltendefinitionen. Die einzelnen Definitionen sind durch Kommas zu trennen. Am Satzende darf kein Komma angegeben werden. Die maximale Länge von 250 Bytes pro Satz sollte nicht überschritten werden.

Beispiele:

```
AD :21:30
```

Von den Datensätzen der Display-Datei wird nur der Teil ab Spalte 21 in der Länge von 30 Bytes angezeigt. Bei Datensätzen mit einer kleineren Länge wird die Darstellung der Spalte entsprechend verkürzt.

```
AD :21-40:"<-- SPALTE 21-40 -->",:1:20"<-- SPALTE 1-20 ---
>",4X'FF'
```

Die Darstellung beginnt mit Spalte 21 bis 40 der in der Display-Datei gespeicherten Sätze. Entsprechendes wird auch in der Scale-Zeile angezeigt ("..."). Falls Datensätze kürzer als 50 Bytes sind, so wird der fehlende Bereich in der Darstellung mit Blanks (C' ') aufgefüllt. Mit dem zweiten, im AD-Kommando angegebenen Parameter wird Spalte 1 bis 20 der Datensätze ausgewählt. Der dritte Parameter bewirkt, daß die Satzdarstellung im Display mit der Konstanten X'FFFFFFFF' abgeschlossen wird.

```
AD (AUSW)
```

Von den Datensätzen der Display-Datei werden die in der Datei AUSW hinterlegten Spaltenbereiche ausgewählt und am Bildschirm dargestellt.
Die Datei AUSW habe folgenden Inhalt:

```
:35-38:"TSN",' '
:45:8"PROGRAMM",' '
:30:10"ISAM-KEY",' '
........
:100-$:"SATZ-REST"
```

Auf den Bereich von Spalte 35 bis 38 folgt ein Blank. Danach Spalte 45 bis 52 und wieder ein Blank usw. Die ausgewählten Daten werden durch entsprechende Bezeichnungen in der Scale-Zeile beschrieben.

Hinweise:

Falls auf eine bereits im AD-Modus dargestellte Datei ein neues AD-Kommando angewendet wird, so ist zu beachten, daß sich die Spaltenangaben auf das Originalformat der Datensätze und nicht auf das durch AD dargestellte Satzformat beziehen.

Der AD-Modus gilt jeweils nur für die im Display befindliche Datei. Bei einem Wechsel zur nächsten Display-Datei mit dem Kommando NF (Next File) bzw. D wird der AD-Modus ausgeschaltet. Durch das Kommando AD ohne Operanden kann die zuletzt definierte Spaltenauswahl jederzeit wieder aktiviert werden.

Bei der Angabe eines Spaltenbereichs in der Form *:col1-col2:* kann für col2 auch das Zeichen '$' angegeben werden. $ steht hierbei für die jeweils letzte Spalte des Datensatzes. Durch die Angabe :101-$: wird z.B. der Bereich von Spalte 101 bis zum Satzende ausgewählt. Eine Spaltendefinition der Art *:col1-$:* kann nur als letzte angegeben werden.

Wird durch *:col1-col2:* bzw. *:col1:len* ein Spaltenbereich definiert, der länger als ein bestimmter Datensatz ist, so sind zwei Fälle zu unterscheiden:

a) Der angegebene Spaltenbereich ist nicht der letzte.
Die im Datensatz fehlenden Stellen werden durch ein festgelegtes Füllzeichen (Standard: C' ') aufgefüllt.

b) Der angegebene Spaltenbereich ist der letzte in der Definition.
Der Spaltenbereich wird nur in der Länge der im Satz vorhandenen Daten ausgegeben.

Im AD-Kommando können Spalten auch mehrfach ausgewählt werden. Die Länge der angezeigten Daten kann dadurch auf ein Vielfaches der ursprünglichen Satzlänge an-

wachsen. Es ist jedoch zu beachten, daß die maximale Länge von 32000 Stellen nicht überschritten wird.

Ist die Display-Datei vom Fcbtyp ISAM, so ist zu beachten, daß in der Spaltenauswahl ein Bereich enthalten sein muß, der den ISAM-Schlüssel oder einen Teil vom Beginn des ISAM-Schlüssels umfaßt.

Die durch das AD-Kommando ausgewählten und am Bildschirm dargestellten Daten werden wie normale Sätze einer Datei behandelt. Im AD-Modus sind alle Display-Kommandos, wie DS/DW/HEX/R/L/C sowie M (Modify) und S (Suchen) erlaubt. Mit dem Kommando W (Write) werden die Datensätze mit dem im Display dargestellten Format in eine andere Datei übertragen. Bei ISAM-Dateien wird dabei auch die aktuell im Display geltende Position des ISAM-Schlüssels (s.o.) übernommen.

Falls eine Spalte mehrfach ausgewählt und im Modify-Modus Daten geändert wurde, so ist zu beachten, daß die Änderung nur dann in die Datei zurückgeschrieben wird, falls sie in der letzten, d.h. in der am weitesten rechts vorkommenden Auswahl der Spalte vorgenommen wurde.

Die Kommandos AD und W in Kombination können dazu benutzt werden, eine Datei zu erstellen, bei der gegenüber dem Original zwei Spaltenbereiche vertauscht sind. Durch das unten angegebene AD-Kommando werden in der Darstellung die Spalten 21-30 und 41-50 getauscht. Mit dem Write-Kommando werden die dargestellten Daten in die Datei CFS.SP.TAUSCH geschrieben.

```
AD :1-20:,:41-50:,:31-40:,:21-30:,:51-$:
W9999,CFS.TAUSCH
```

Handelte es sich bei der ursprünglichen Datei um eine ISAM-Datei mit KEYPOS=25, KEYLEN=8 und RECFORM=V, so wird CFS.TAUSCH ebenfalls als ISAM-Datei, jedoch mit KEYPOS=45, KEYLEN=8 angelegt. Eine Verschiebung des ISAM-Schlüssels wird beim Anlegen einer neuen Datei durch das Write-Kommando berücksichtigt.

Durch die H-Option bei der Definition der Spalten ist es möglich, in der gleichen Bildschirmdarstellung hexadezimale und Character-Daten zu mischen. Wird bei einer mit dem Zusatz H bzw. X versehenen Spaltendefinition die Hexadezimaldarstellung aktiviert, so können die Daten dieser Spalte im Modify-Modus auch hexadezimal verändert werden. Andere Spalten, die ohne den Zusatz H definiert wurden, können im gleichen Eingabeschritt im Character-Modus geändert werden.

Display-Datei nach jeder Maskeneingabe schließen

CA | NCA

Die Display-Datei wird nach jeder Maskeneingabe geschlossen. Während der Eingabepausen am Bildschirm ist die Datei geschlossen und kann von anderen Benutzern lesend oder schreibend eröffnet werden.
Standard: NCA

Hinweise:

Beim Modifizieren von Dateien (Modify) bleibt der CA-Modus ohne Wirkung. D.h. die zur Änderung freigegebene Datei bleibt bis zum Verlassen des Modify-Modus gegen Schreibzugriff durch andere Benutzer geschützt. Der CA-Modus ist ebenfalls ohne Wirkung bei der Einzelsatzdarstellung (Single Record-Modus, SR), sowie beim Display von Banddateien.

Der CA-Modus wird mit dem Verlassen des Display-Modus nicht ausgeschaltet, sondern wirkt auch für alle folgenden Display-Aktionen solange bis er durch das Kommando NCA wieder zurückgenommen wird.

Nächste Display-Datei anzeigen

D

Display Next File. Das nächste mit dem Action-Code D oder M markierte Datenobjekt wird angezeigt. Anstelle von D kann auch NF angegeben werden.

Art des Datenzugriffs auf Datei festlegen

DA *typ* [, *param*]

Mit diesem Kommando kann ein alternativer Datenzugriff auf die aktuell im Display befindliche Datei festgelegt werden. Dies bedeutet, daß die Datensätze in einer anderen Sortierreihenfolge bzw. in einem anderen Datenformat angezeigt werden.

typ

Art des Datenzugriffs. Gegenwärtig sind folgende Zugriffsarten realisiert:

I Zugriff auf eine NK-ISAM Datei gemäß einem mit CREate-Alternate-Index eingerichteten Sekundärschlüssel.

CODE Zeichenweises Konvertieren der Datensätze/Datenblöcke für die Bildschirmdarstellung. Die Datensätze in der Datei werden nicht verändert.

FLAM Zugriff auf eine mit dem Softwareprodukt FLAM komprimierte Datei. Die Datensätze werden im Originalformat, d.h. unkomprimiert angezeigt.

param

Zusatzparameter für den Datenzugriff. Je nach Art des Datenzugriffs (CONV/I/FLAM) hat der Parameter verschiedene Bedeutungen:

CODE {, UPPER | UPPER2 | LOWER | EBCTOASC | ASCTOEBC | *datei*} [, K=Y]

UPPER Umsetzung aller Kleinbuchstaben in Großbuchstaben.
X'00' --> X'00' (unverändert)
X'4F' (ö bzw. |) --> X'4F' (unverändert)
X'FB' (ä bzw. {) --> X'FB' (unverändert)
X'FD' (ü bzw. }) --> X'FD' (unverändert)

UPPER2 Umsetzung aller Kleinbuchstaben in Großbuchstaben.
X'00' --> X'40' (Blank)
X'4F' (ö bzw. |) --> X'BC' (Ö bzw. ~)
X'FB' (ä bzw. {) --> X'BB' (Ä bzw. [)
X'FD' (ü bzw. }) --> X'BD' (Ü bzw.])

LOWER Umsetzung aller Großbuchstaben in Kleinbuchstaben.
X'40' --> X'40' (unverändert)
X'BC' (Ö bzw. ~) --> X'BC' (unverändert)
X'BB' (Ä bzw. [) --> X'BB' (unverändert)
X'BD' (Ü bzw.]) --> X'BD' (unverändert)

ASCTOEBC Umsetzung von ASCII Zeichen nach EBCDIC.

EBCTOASC Umsetzung von EBCDIC Zeichen nach ASCII.

datei Umsetzung gemäß einer benutzereigenen Tabelle. Die Umsetztabelle ist in der angegebenen Datei (SAM, Recform=V) definiert. Die Datei enthält 16 Datensätze entsprechend den umzusetzenden Codepositionen 00 - 0F (Satz 1), 10 - 1F (Satz 2), ..., F0 - FF (Satz 16). Jeder Datensatz ist 16 oder 32 Byte lang. Bei 32 Byte werden die Daten als hexadezimale

Codierungen interpretiert. Bei 16 Byte werden die Daten als direkte Character Codierungen interpretiert.

K=Y Bei ISAM-Dateien sollen auch die Daten des Schlüssels konvertiert werden. Standardmäßig werden die ISAM-Schlüssel von der Konvertierung ausgenommen.

Eine neue Datei, die tatsächlich die umgesetzten Zeichen enthält, kann mit dem Kommando W$,*datei* erzeugt werden.

I Beim ISAM-Zugriff wird mit diesem Parameter der gewünschte Alternate Index angegeben. Die Datensätze werden aufsteigend gemäß dem Alternate Index angezeigt.

FLAM Bei der Option FLAM können Parameter angegeben werden, die die Art der Dekomprimierung beeinflussen, z.B. DAFLAM,TR=A/E. Durch diesen Parameter werden die Daten bei der Dekomprimierung vom ASCII- in den EBCDIC-Zeichensatz umgesetzt. Für eine ausführliche Beschreibung aller Parameter siehe "FLAM-Benutzerhandbuch".

NDA Die Datenzugriffsart wird auf die Standardeinstellung zurückgesetzt.

Hinweise:

Mit dem DA-Kommando können auch benutzerspezifische Zugriffe auf Datenbanken realisiert werden. Eine detaillierte Beschreibung des Data Access Interface ist auf Wunsch verfügbar.

Durch das W-Kommando können die Datensätze in der dargestellten Reihenfolge bzw. in dem dargestellten Format (d.h. bei FLAM entkomprimiert) in eine andere Datei geschrieben werden. Bei DA I,*key* wird in der durch Write erzeugten ISAM-Datei die Keypos und Keylen des gerade aktivierten Sekundärschlüssels verwendet.

Ein durch das Kommando DA festgelegter alternativer Datenzugriff auf die Display-Datei gilt solange, bis der Display-Modus durch die K1-Taste bzw. die Kommandos D/NF/LF/LST für die aktuelle Datei beendet wird. Während des Display kann durch das Kommando NDA in die Standardzugriffsmethode umgeschaltet werden.

Display (Edit) Long

DL [,NC] | NDL Display Long/No Display Long.

DL [, NC] Jeder Datensatz wird in seiner ganzen Länge bzw. bis Bildschirmende angezeigt.
Anstelle von DL können auch die Kommandos EL (Edit Long) oder DW (Display Wide) angegeben werden.

NC No Columns. Bei Ausführung eines Suche-Kommandos im DL-Modus wird die Spaltenpositionierung auf den jeweiligen Treffer unterdrückt (gleiches Verhalten wie im DS-Modus). Im gleichen Zusammenhang siehe auch Seite **310**: Parameter NC im Suche-Kommando.

NDL Es wird von jedem Datensatz nur soviel angezeigt, wie in eine Bildschirmzeile paßt.
Anstelle von NDL können auch die Kommandos ELO (Edit Long Off) oder DS (Display Short) angegeben werden.
Standard: NDL.

Hinweis:

Als Alternative zur Dateidarstellung im DL-Modus sei auf das Kommando AD (Arrange Data, Seite 297) hingewiesen. Dieses Kommando ermöglicht die selektive Anzeige bestimmter Spaltenbereiche einer Datei. Die Reihenfolge der Spaltenbereiche und deren Länge ist frei bestimmbar. Für jeden ausgewählten Spaltenbereich kann ein Darstellungsformat (Character/Hexadezimal) gewählt werden.

Hexadezimale Darstellung

H | NH

Hexadezimal display/No Hexadezimal display.
Anstelle von H/NH können auch die Kommandos HEX/HEXO angegeben werden.

Standard: NH.

Hexadezimale Spaltendarstellung

HEXC | NHEXC

Hexadezimale Spaltendarstellung/dezimale Spaltendarstellung.

Bei Hexadezimal-Darstellung erfolgt die Anzeige der Spaltenposition im EL-Modus (Edit Long) am linken Rand des Bildschirms in dezimalen Zahlen, z.B. (0025) Durch das Kommando HEXC werden die Spalten in hexadezimalen Werten angezeigt, z.B. (0018). Im Dezimalmodus ist 1 die erste Spalte. Im Hexadezimalmodus beginnt die Spaltenzählung mit 0. Siehe hierzu auch Kommando CX'..'.

Standard: NHEXC.

Anzeigemodus festhalten

KDO | NKDO

Keep Display Options/do Not Keep Display Options.

Im Normalfall (NKDO) werden die verschiedenen Optionen der Darstellung von Daten beim Beenden des Display-Modus bzw. beim Übergang zur Anzeige einer anderen Display-Datei auf die Standardwerte zurückgesetzt.
Das Kommando KDO bewirkt, daß die aktuellen Werte der folgenden Parameter festgehalten und beim Anzeigen einer neuen Datei im Display-Modus automatisch aktiviert werden:

C*nn*	erste angezeigte Spalte
DL/DS	Display Long/Display Short
H/NH	Hexadezimale/Character-Darstellung
N/NN	Satznummern anzeigen/nicht anzeigen
OL/NOL	Zeilenlineal anzeigen/nicht anzeigen
SR/NSR [SHARUPD=YES]	Einzelsatzdarstellung/SHARUPD=YES

Standard: NKDO.

Hinweis:

Die gewünschten Display-Modi können bereits beim Start von CFS über eine Startup-Datei (siehe Seite 537) eingestellt und für den aktuellen CFS-Lauf als Standardwerte festgehalten werden. Die Startup-Datei könnte z.B. folgendes Aussehen haben:

```
*002 FILE=NO
NSR SHARUPD=YES
DL
N
KDO
*DIA
```

Die mit KDO festgehaltenen Display-Optionen gelten bis zur Beendigung des Programms CFS.

Letzte Display-Datei wieder anzeigen

LF Last File. Das vorhergehende Display-Datenobjekt wird wieder angezeigt.

Aus dem Display zur Anzeige der Dateienliste zurückkehren

LST LiSTe. Wieder zur Anzeige der Dateienliste zurückkehren.

Hinweis:

Anstelle des Kommandos LST kann auch die K1-/F1-Taste zur Rückkehr in die Dateienliste verwendet werden. Die F1-Taste hat dabei den Zusatzeffekt, daß Änderungen im Modify-Modus in die Datei zurückgeschrieben werden, selbst wenn diese im gleichen Dialogschritt wie die Betätigung der F1-Taste vorgenommen wurden.

Anzeige der Satz-/Blocknummern

N | NN record Numbers/No record Numbers.

Bei SAM-Dateien werden die Satznummern angezeigt/nicht angezeigt.
Bei PAM-Dateien werden die Blocknummern angezeigt/nicht angezeigt.

Standard: N (Satz-/Blocknummern werden angezeigt).

Nächste Display-Datei anzeigen

NF Next File. Das nächste mit dem Action-Code D oder M bezeichnete Datenobjekt wird angezeigt. Anstelle von NF kann auch D angegeben werden.

Hinweis:

Hat der Benutzer in einer Dateienliste mehrere Datenobjekte mit D bzw. M gekennzeichnet, er möchte diese jedoch nicht in der Reihenfolge der markierten Dateien innerhalb der Liste angezeigt bekommen, sondern z.B. die dritte Datei zuerst, so kann er gleichzeitig mit dem Senden der Action-Codes das Kommando NF mehrmals verknüpfen (z.B.: D;NF;NF) und erhält dann die der Anzahl des Kommandos NF (oder D) entsprechende Datei im Display.

Satzanfang für Display festlegen/ändern

Standardmäßig werden die Sätze von RECFORM=V Dateien im CFS-Display mit dem ersten Byte nach dem 4-stelligen Satzlängenfeld angezeigt. Mit dem Offset Kommando O kann diese Standardeinstellung für die gerade angezeigte Datei geändert werden.

O [*x*] *n*	Spalte +/-*n* wird in Zukunft als logischer Satzanfang betrachtet.
x	+ | - (Standard: +)
n	0 - 999
O 0	Der Offset für die Dateianzeige wird auf den Standardwert zurückgesetzt.

Beispiele:

`O-4`
Alle Datensätze einer RECFORM=V Datei werden einschließlich Satzlängenfeld angezeigt. Die Spaltenangabe C1 (Column 1) bezieht sich in diesem Fall auf den Beginn des Satzlängenfeldes und nicht auf den Beginn der Daten.
Mit S,:1:>X'0054' können nach Eingabe von O-4 alle Datensätze gesucht werden, die mehr als 80 Datenbytes enthalten (Satzlänge > 80 + 4).

`O+8`
Bei ISAM-Dateien mit KEYPOS=5,KEYLEN=8 werden die ISAM-Schlüssel nicht angezeigt. Die Spaltenangabe C1 (Column 1) bezieht sich in diesem Fall auf die Stelle nach dem ISAM-Schlüssel und nicht auf den Beginn des ISAM-Schlüssels, wie dies standardmäßig der Fall ist.

Sichtfenster auf Satz/Datenblock positionieren

P*n*

SAM/PAM: Sichtfenster auf Satz/Block mit der angegebenen Nummer positionieren. Anstelle von P*n* kann auch das Kommando #*n* angegeben werden.

Hinweis:

Bei PAM-Dateien kann nur bis zum letzten beschriebenen Datenblock positioniert werden. Nur unter TSOS werden auch die der Datei zugewiesenen, jedoch noch nicht beschriebenen PAM-Blöcke im Display angezeigt.

P*key* | PX'*hex*'

ISAM: Sichtfenster auf Satz mit dem angegebenen ISAM-Schlüssel positionieren.

Anstelle von P*key*/PX'*key*' kann auch #*key*/#X'*key*' angegeben werden.

PB*nnn*

Bei SAM-Dateien: Position to Buffer.
Bei PAM-Dateien mit BLKSIZE=(STD,1): Position to PAM-Block.
Bei PAM-Dateien mit BLKSIZE > (STD,1): Position to logical PAM-Buffer.
Das Sichtfenster wird auf den logischen Puffer mit der angegebenen Nummer positioniert. Die Anzeige beginnt mit dem ersten Satz im angegebenen Puffer. Die Größe des Datenpuffers bei SAM-Dateien hängt ab von der Blocksize der Datei und ist ein Vielfaches von 2048 Bytes. Anstelle von PB*nnn* kann auch #B*nnn* angegeben werden.

Hinweise:

Das PB-Kommando bietet bei sehr großen SAM-Dateien die Möglichkeit, das Sichtfenster bedeutend schneller als mit dem normalen Kommando ++ an das Dateiende zu positionieren, da bei PB direkt der entsprechende Block gelesen wird.

Beim Positionieren auf SAM-Puffer verliert CFS die Kontrolle über die Satznummern. Es wird daher automatisch auf den NN-Modus (No Numbers) umgeschaltet.

PP*nnn* Position to Pam-Block.
Falls die Datei eine BLKSIZE = (STD,*b*) besitzt, so wird das Sichtfenster durch PP*nnn* auf den PAM-Block INTEGER(*nnn*/*b*) positioniert.

Anzeigeposition merken/Dateiausschnitt an vorher gemerkte Positi-on verschieben

PD.*n* Position Define. Die Position des angezeigten Dateiausschnitts wird zum Zwecke des späteren Zurückpositionierens gemerkt.

n Ziffer zwischen 0 und 9.

P.*n* Positionieren an definierte Stelle: Das Display springt an die Stelle in der Datei, die zuvor mit einem PD-Kommando definiert wurde.

n Ziffer zwischen 0 und 9.

SAM-/ISAM-Datei im PAM-Format anzeigen

PAM | NPAM PAM-Format/No PAM-Format.

PAM SAM-/ISAM-Dateien werden im PAM-Format dargestellt.

NPAM Der PAM-Modus für SAM-/ISAM-Dateien wird ausgeschaltet.
Standard: NPAM.

Hinweis:

PAM-Dateien werden stets im DL-Modus (Display Long) angezeigt.
Falls sich der Benutzer nur für einen bestimmten Teil der Sätze einer PAM-Datei interessiert (z.B. die ersten 80 Bytes eines jeden PAM-Blockes), so kann dies mit dem Kommando AD (Arrange Data) erreicht werden

Anzeige der PAM-Keys

PK | NPK PAM-Keys/No PAM-Keys.

Bei PAM-Dateien / bei der Anzeige von SAM-/ISAM-Dateien im PAM-Format werden anstelle der Datenblöcke die 16 Byte langen PAM-Keys angezeigt.
Auf die PAM-Keys sind alle Display-Kommandos, wie z.B. HEX/HEXO/M/NM/S/W anwendbar.

Standard: NPK

Spaltenzähler einblenden

SC [*datei* | 'text']

SC In der oberen Bildschirmhälfte wird ein "Lineal" mit einem Spaltenzähler angezeigt. Das Lineal ist im Display-Modus und bei der Anzeige der Dateienliste aktiv.

SC *datei* Das Lineal besteht aus dem ersten Satz der angegebenen Datei. Bei Anzeige der Dateienliste wird das Lineal in der Standardform, d.h. ohne Satzinhalt dargestellt.

SC '*text*' Das Lineal besteht aus dem in Hochkommas angegebenen Text.

SCO | NSC Das eingeblendete Lineal wird ausgeschaltet. Anstelle von SCO kann auch NSC angegeben werden. Standard: SCO.

OL | NOL Alternativ zu SC/SCO können auch die Kommandos OL/NOL (Orientation Line/No Orientation Line) angegeben werden.

Hinweise:

Beim Display von ISAM-Dateien werden die Stellen des ISAM-Schlüssels im Lineal je nach Bildschirmtyp hell oder unterstrichen dargestellt.

Scale ist nützlich als Orientierungshilfe für das Suche-Kommando (Angabe eines Spaltenbereichs). Das Auffinden bestimmter Spalten innerhalb der Datensätze wird durch Scale ebenfalls erleichtert.

Single Record-Modus (Sharupd=Yes) im CFS-Display/Editor

SR | NSR [SHARUPD=YES]

Single Record Modus / No Single Record Modus.

SR Der Single Record-Modus bezieht sich ausschließlich auf den Display/Modify-Modus von CFS und hat zwei verschiedene Wirkungen:

a) Es wird jeweils nur ein Satz bzw. Datenblock am Bildschirm angezeigt.

b) Bei ISAM- und PAM-Dateien wird in den Sharupd-Modus gewechselt.

Der SR-Modus garantiert die Integrität der Daten auch bei konkurrierendem Zugriff durch verschiedene Benutzer.

NSR Durch das Kommando NSR wird der Single Record-Modus ausgeschaltet. Eine im Display befindliche ISAM-Datei wird mit SHARUPD=NO eröffnet.

NSR SHARUPD=YES Der Zusatz SHARUPD=YES im NSR-Kommando bewirkt, daß die Display-Datei auch im Modus der ganzseitigen Anzeige mit SHARUPD=YES eröffnet ist.

Standard: NSR. ISAM-Dateien werden mit SHARUPD=NO angezeigt und modifiziert.

Hinweise:

Der SR-Modus gilt nur für die aktuelle Display-Datei und muß bei einer neuen Datei gegebenenfalls wieder gesetzt werden.

Durch die Kommandofolge SR;KDO bzw. NSR SHARUPD=YES;KDO kann SHARUPD= YES als Standard für alle folgenden Display-Operationen festgesetzt werden. Für weitere Informationen zum Kommando KDO siehe Seite 304.

Der SR-Modus ist auch beim Display von Banddateien nützlich. Es entfällt damit das Rückspulen des Bandes vor jeder Maskenausgabe.

Suchen von Zeichenfolgen (einfaches Suchargument)

S [-] [*n*] ,[*col*] [*r*] *item* [, R] [, NC] [, SR]

Vom ersten im Sichtfenster angezeigten Satz bis Dateiende/Dateianfang wird nach der angegebenen Zeichenfolge gesucht. Das Sichtfenster wird auf den Satz positioniert, der den ersten Treffer gebracht hat. Im Kommandofeld wird ein Suche-Kommando zum Auffinden des nächsten Treffers vorgegeben.
Durch Drücken der ENTER-Taste (Absenden des Eingabevorschlags) wird die Suche fortgesetzt.

\- Rückwärtssuche: Die Suche erfolgt vom ersten im Sichtfenster angezeigten Datensatz in Richtung Dateianfang.
Standard: Suche in Richtung Dateiende.

n Anzahl der Sätze, in denen nach dem Suchargument gesucht wird.
Standard: unbegrenzt viele Sätze.

col Spaltenbereich in dem die gesuchte Zeichenfolge beginnen muß.

:col1-col2:
Das erste Zeichen der gesuchten Zeichenfolge muß im Spaltenbereich zwischen col1 und col2 beginnen.

:col1:
Die Zeichenfolge wird nur an der Spalte col1 gesucht und muß dort beginnen.

>:col1: | <:col1:
Die Zeichenfolge wird im Bereich ab Spalte col1 bis Satzende (>:col1:) bzw. vom Satzanfang bis Spalte col1 gesucht (<:col1:).

Standard: Suche in gesamten Spaltenbereich (von Spalte 1 bis Satzende).

r > | < | -
\> Suche nach einer Zeichenfolge > *item*
< Suche nach einer Zeichenfolge < *item*
\- Suche nach einer Zeichenfolge ungleich *item*
Standard: Suche nach einer Zeichenfolge = *item*.

item Suchzeichenfolge: C'*string*' | L'*string*' | X'*string*' | A'*string*'

C'*string*' kann zu '*string*' abgekürzt werden. In vielen Fällen können sogar die Hochkommas weggelassen werden, siehe Hinweise.
Enthält *string* Hochkommas ('), so müssen diese verdoppelt werden ('').
Die maximale Länge von *string* beträgt 128 Bytes.

L'*string*': Kleinbuchstaben in *string* werden nicht in Großbuchstaben umgesetzt.

A'*string*': Bei der Suche werden Klein- und Großbuchstaben gleich behandelt.
Beispiel: Der Suchbegriff A'CFS' wird als Treffer erkannt, falls im Datensatz die Zeichenfolge 'CFS' oder 'cfs' oder eine beliebige Kombination aus Groß- und Kleinbuchstaben enthalten ist.
Über das Kommando SET SEARCHMODE=A kann der A-Modus voreingestellt werden. S,'*string*' wirkt dann wie S,A'*string*'. Die Rückkehr in den Standardmodus erfolgt durch SET SEARCHMODE=C.

Das Suche-Kommando bietet auch die Möglichkeit, mehrere Suchargumente mit Und-, Oder- bzw. Wildcard-Syntax zu verbinden. Die Syntax ist eine Aneinanderreihung einfacher Suchargumente und wird im nächsten Abschnitt "Suchen von Zeichenfolgen (mehrere Suchargumente)" ausführlich beschrieben.

R — Reverse. Die Suche innerhalb der Datensätze erfolgt nicht von links nach rechts, sondern in umgekehrter Reihenfolge von rechts nach links. In diesem Modus können keine Und-, Oder-, Wildcard-Verknüpfungen von mehreren Suchbegriffen angegeben werden.

NC — No Columns. Im DL-Modus (Display Long) wird bei der aktuellen Suche die Spaltenpositionierung auf den jeweiligen Treffer unterdrückt (gleiches Verhalten wie im DS-Modus). Im gleichen Zusammenhang ist auch der optionale Parameter NC im Kommando DL von Bedeutung (siehe Seite **303**). Durch DL,NC wird der NC-Modus für alle folgenden Suche-Kommandos in der aktuellen Datei eingestellt.

SR — Single Record. Dieser Indikator ist nur von Bedeutung bei einer Negativ-Suche, d.h. bei der Suche nach einer Zeichenfolge ungleich einem Wert (z.B. S,-X'00').
Standardmäßig wird bei einer Negativ-Suche ohne Spaltenbereich ein Satz nur dann als Treffer angesehen, falls der Suchstring im gesamten Satz nicht vorkommt.
Im SR-Modus wird das erste Nicht-Vorkommen des Strings im Satz als Treffer gewertet. Bei der Angabe der Option R (siehe oben) wird automatisch der SR-Modus eingeschaltet.

S [-] — Suche vom ersten angezeigten Satz bis Dateiende/Dateianfang nach dem zuletzt definierten Suchargument.

S? | S?,%*name*

Das zuletzt definierte Suchargument/das mit SP %*name* in einem Parameter-Set gespeicherte Suchargument wird angezeigt. Beschreibung von SP %*name* siehe Seite **315**.

Hinweise:

Bei der Angabe des **Spaltenbereichs** *col* in der Suchanweisung sind einige Unterschiede zur Spaltenbereichsangabe im EDT zu beachten:

1) :col1-col2: besagt, daß der in Hochkommas eingeschlossene Suchbegriff in dem angegebenen Spaltenbereich beginnen muß.
Im EDT muß der Suchbegriff vollständig im angegebenen Spaltenbereich enthalten sein.

2) :col: besagt, daß der in Hochkommas eingeschlossene Suchbegriff genau in der angegebenen Spalte beginnen muß.
Im EDT bewirkt die Angabe einer einzigen Spalte :col:, daß der Suchbegriff von Spalte :col: bis Satzende gesucht wird. In CFS wird dies durch die Angabe >:col2: erreicht, wobei col2=col-1 ist. Durch <:col: kann in CFS eine Zeichenfolge im Bereich vom Satzanfang bis zur Spalte col gesucht werden.

'*string*' kann in den meisten Fällen auch ohne Hochkommas angegeben werden (S,*string*). Die Hochkommas dürfen lediglich in den Fällen nicht weggelassen werden, in denen *string* Leerzeichen enthält bzw. von einem Ersatzstring (Seite **317**) oder Write-Kommando (Seite **322**) gefolgt wird.

Das einmal definierte Suchargument gilt für alle folgenden S-Kommandos und braucht im folgenden nicht mehr angegeben zu werden. Dies gilt solange, bis eine neue Zeichenfolge als Suchbegriff verwendet wird.

Ist der Benutzer nur an der **Anzahl der Treffer** in der Datei interessiert, so kann er dies folgendermaßen erreichen: Suche-Kommando S,'*string*'='',A angeben (nur im Non-Modify-Modus). In diesem Fall wird nur die Anzahl der gefundenen Treffer bis Dateiende gezählt.

Während der Ausführung eines langdauernden Suche-Kommandos kann durch Anwendung der K2-Taste (Verzweigen in die BS2000 Kommando-Ebene) und anschlie-

ßendes /INTR R die begonnene Suche abgebrochen werden. Das Sichtfenster wird in diesem Fall auf die zuletzt gelesene Stelle in der Datei positioniert.

PAM: **Suchen über Blockgrenzen**

Bei PAM-Dateien mit BLKSIZE=(STD,1) muß der zu suchende String nicht vollständig in einem Datenblock enthalten sein. Falls der Suchbegriff (z.B. 'DATABASE') am Ende eines PAM-Blocks beginnt und sich am Anfang des nächsten PAM-Blocks fortsetzt (z.B. 'DA' + 'TABASE'), so wird diese Konstellation von CFS ebenfalls als Treffer erkannt.

Beispiele:

`S,C'XYZ'`
Suche ab dem ersten im Display angezeigten Satz bis zum Dateiende die Zeichenfolge C'XYZ'.

`S,C'C'' '''`
Suche nach der Zeichenfolge C'␣' in der gesamten Datei (Hochkommas in dem zu suchenden String müssen verdoppelt werden).

`S,L'Datei'`
Suche alle Datensätze, die die Zeichenfolge 'Datei' enthalten. Die im String enthaltenen Kleinbuchstaben werden vor der Suche nicht umgesetzt.

`S,A'datei'`
Suche alle Datensätze, die die Zeichenfolge 'datei' enthalten. Bei der Suche ist die Groß-/Kleinschreibung der Zeichenfolge im Datensatz unerheblich.

`S,:73:-' '`
Suche in Spalte 73 eines jeden Satzes nach einem Zeichen ungleich Blank. Sätze mit weniger als 73 Stellen werden hier nicht als Treffer erkannt.

`S,:1:>X'0054'`
Zur Ermittlung von Datensätzen einer bestimmten Länge kann mit dem Offset-Kommando O-4 das Satzlängenfeld in das Display miteinbezogen werden. Durch das angegebene Suche-Kommando werden in einer RECFORM=V Datei alle Datensätze gefunden, deren Länge inclusive Satzlängenfeld größer als 84 Bytes ist.

`S,'CFS'=''`
Im Non-Modify-Modus: Es wird die Anzahl der Datensätze gezählt, in denen der Suchbegriff 'CFS' mindestens einmal vorkommt. Im Non-Modify-Modus wird keine Ersetzung des Suchstrings in der Datei vorgenommen.

`S,'CFS'='',A`
Im Non-Modify-Modus: Es wird die Gesamtanzahl der Treffer gezählt.
A: mehrfache Treffer in einem Datensatz werden entsprechend oft gezählt.
Im Modify-Modus würde der Suchstring in ISAM-Dateien bei jedem Vorkommen innerhalb eines Satzes gelöscht.

`S,''`
Gibt in der Hinweiszeile die Länge des ersten im Sichtfenster angezeigten Satzes aus (sucht nach dem Satzende).

Suchen (Vergleichen von Spaltenbereichen)

S [-] [*n*] ,:*col1*-*col2*: *r* :*col3*:

Vom ersten im Sichtfenster angezeigten Satz bis Dateiende/Dateianfang wird der Inhalt des mit :*col1*-*col2*: angegebenen Spaltenbereichs mit dem Inhalt des bei :*col3*: beginnenden Spaltenbereichs verglichen. Das Sichtfenster wird auf den ersten Satz positioniert, bei dem die beiden Spaltenbereiche im Inhalt übereinstimmen. Im Kom-

mandofeld wird ein Suche-Kommando zum Auffinden des nächsten Treffers vorgegeben. Durch Drücken der ENTER-Taste (Absenden des Eingabevorschlags) wird die Suche fortgesetzt.

- Rückwärtssuche: Die Suche erfolgt vom ersten im Sichtfenster angezeigten Datensatz in Richtung Dateianfang.
  Standard: Suche in Richtung Dateiende.

n Anzahl der Sätze, in denen gesucht werden soll.
Standard: unbegrenzt viele Sätze.

col1 Anfangsspalte des ersten Bereichs.

col2 Endespalte des ersten Bereichs.

r = | > | < | -

= die Spaltenbereiche müssen im Inhalt übereinstimmen.
> der Inhalt des ersten Spaltenbereichs muß größer als der Inhalt des zweiten Spaltenbereichs sein.
< der Inhalt des ersten Spaltenbereichs muß kleiner als der Inhalt des zweiten Spaltenbereichs sein.
-= der Inhalt des ersten Spaltenbereichs muß ungleich dem Inhalt des zweiten Spaltenbereichs sein.

Standard: =

Hinweise:

Beide Spaltenbereiche müssen vollständig im Satz enthalten sein. Falls einer der Bereiche über das Satzende hinausgeht, wird von CFS Ungleichheit signalisiert.

Das Vergleichen von Spaltenbereichen kann wie das Suchen von Zeichenfolgen mit den Operatoren "+" und "," mit weiteren Suchbedingungen verknüpft werden. Näheres hierzu im Abschnitt "Suchen von Zeichenfolgen (mehrere Suchargumente)".

Beispiele:

`S,:1-20:=:101:`
Es wird der Inhalt der ersten 20 Bytes jedes Satzes mit dem Inhalt der Spalte 101 bis 120 verglichen. Bei Gleichheit wird ein Treffer signalisiert.

`S,:1-1:>:12:`
Es wird der Inhalt der ersten Spalte jedes Satzes mit dem Inhalt der Spalte 12 verglichen. Falls der Inhalt von Spalte 1 größer als der Inhalt von Spalte 12 ist, wird ein Treffer signalisiert.

Suchen von Zeichenfolgen (mehrere Suchargumente)

S [-] [*n*] , *such* [*vk such*] [...] ..... [, A] [, NC] Format 1

S ... , ( ) [, A] [, NC] Format 2

S ... , *%name* | %? [, A] [, NC] Format 3

S ... , (*s-dat*) [, A] [, NC] Format 4

Vom ersten im Sichtfenster angezeigten Satz bis Dateiende/Dateianfang wird nach der Kombination der angegebenen mehrfachen Suchargumente gesucht.

Die Syntax für die einzelnen Suchargumente *such* ist im Abschnitt "Suchen von Zeichenfolgen (einfaches Suchargument)" beschrieben.
Jedes Suchargument wird durch einen Operator *vk* mit dem jeweils nächsten Suchargument verknüpft. Die Anzahl der zu verknüpfenden Suchargumente ist beliebig.

In der obigen und in den folgenden Syntaxbeschreibungen steht S [-] [*n*] für den Bereich, über den sich die Suche erstrecken soll: Rückwärtssuche, Anzahl der Sätze, in denen gesucht werden soll. Ausführliche Beschreibung Seite **309**.

Im folgenden werden die 4 Formate der Mehrfachsuche beschrieben.

Mehrfachsuche - Format 1

S [-] [*n*] , *such* [*vk such*] [...] ..... [, A]

such [*col*] [*r*] *item*
einfaches Suchargument wie im vorhergehenden Abschnitt "Suchen von Zeichenfolgen (einfaches Suchargument)" ausführlich beschrieben.

vk , | + | *
Verknüpfungsoperator mit dem vorausgegangenen einfachen Suchargument *such*.

, Suche im aktuellen Satz das vorausgegangene <u>oder</u> das nachfolgende Suchargument. Die Suchbedingung gilt als erfüllt, wenn zumindest eines der beiden Such-Items im Datensatz enthalten ist.

+ Suche im aktuellen Satz das vorausgegangene <u>und</u> das nachfolgende Suchargument. Die Suchbedingung ist erfüllt, wenn beide Suchargumente im Datensatz enthalten sind. Die Reihenfolge der Suchargumente im Datensatz ist ohne Bedeutung.

*****[*n*] Suche im aktuellen Satz das vorausgegangene <u>und</u> das nachfolgende Suchargument. Die Suchbedingung ist erfüllt, wenn beide Suchargumente im Datensatz enthalten sind. Die Suchargumente müssen in der gleichen Reihenfolge auftreten, wie im Suche-Kommando angegeben.

n Anzahl der Trennzeichen mit beliebigem Inhalt, die im Datensatz zwischen dem ersten und zweiten Suchargument enthalten sein müssen. Falls *n* nicht angegeben wurde, so ist jede beliebige Anzahl von Trennzeichen, einschließlich 0 möglich.

Es können beliebig viele Konstrukte der Art *vk such* aneinandergereiht werden.
Bei Verknüpfung durch die Und-Bedingung '+' können auch geklammerte Ausdrücke angegeben werden: (.. , .. , ...)+(.. , .. , ...). Ein Beispiel hierzu ist weiter unten (Seite **315**) beschrieben.

A — All: Bei einer Suche, die mehrere mit "oder" (",") verknüpfte Suchargumente enthält, wird die Suche nicht nach dem ersten im Datensatz gefundenen Item abgebrochen. Dies hat längere Ausführungszeiten beim Suchvorgang zur Folge. Dafür werden jedoch alle Treffer-Items mit den entsprechenden Spaltenangaben in der Hinweiszeile der CFS-Maske angezeigt.
Standard: Die Suche wird nach dem ersten gefundenen Item beendet.

NC — No Columns. Im DL-Modus (Display Long) wird die Spaltenpositionierung auf den jeweiligen Treffer unterdrückt (gleiches Verhalten wie im DS-Modus).

Hinweise:

Die Reihe der Suchargumente und Verknüpfungsoperatoren wird linear abgearbeitet. Falls mehrere mit "+" bzw. "*" verknüpfte Suchargumente angegeben wurden und eines von ihnen nicht im Datensatz enthalten ist, so wird der Suchvorgang beendet bzw. beim nächsten, mit oder "," verknüpften Such-Item fortgesetzt.

Für jedes einzelne Suchargument *such* kann ein Spaltenbereich (:col1-col2: / :col: / >:col: / <:col:), eine Negativ-Bedingung (-'*item*'), sowie ein Such-Item in Normaldarstellung ('*item*'), in hexadezimaler Darstellung (X'*item*') oder ein Such-Item in Groß-/Kleinschreibung (L'*item*') angegeben werden. Bei einem Such-Item in Normaldarstellung können die Hochkommas in der Regel sogar weggelassen werden. Durch S,FLAG*STATEMENT werden z.B. alle Datensätze gesucht, die die Zeichenfolge 'FLAG' und irgendwo danach die Zeichenfolge 'STATEMENT' enthalten. Weitere Informationen siehe Syntaxbeschreibung und Hinweise auf Seite **309**.

In der Kommandozeile von CFS stehen insgesamt 71 Bytes zur Aufnahme des gesamten Mehrfachsuche-Kommandos zur Verfügung. Dies wird sich bei der Angabe einer größeren Anzahl von Suchargumenten oft als Engpaß erweisen. Um dieser Situation zu begegnen, kann über das Kommandofeld von CFS eine spezielle Such-Maske angefordert werden. Zur Anforderung der Suchmaske dient das im folgenden beschriebene Format 2 des Suche-Kommandos: S,( ).

Beispiele:

```
S,'='*'(','DC '*'('*')'
```
Es werden alle Datensätze gesucht, die eine der beiden Bedingungen erfüllen:

- Zeichen '=' und irgendwann danach Zeichen '('. z.B. '=A(...)', '=V(...)'
- Zeichenfolge 'DC ' und irgendwo danach die Zeichen '(' und ')'.
  z.B. 'DC A(...)', 'DC Y(...)'

```
S,:1:>X'004C'+:1:<X'0054'
```
Zur Ermittlung von **Datensätzen einer bestimmten Länge** kann mit dem Offset-Kommando O-4 das Satzlängenfeld in das Display miteinbezogen werden. Durch das angegebene Suche-Kommando werden in einer RECFORM=V Datei alle Datensätze gefunden, deren Länge größer als 76 Bytes und kleiner als 84 Bytes ist.

```
S,-L'a'+-L'b'+-L'c'
```
Es werden alle Sätze gesucht, die keinen der Kleinbuchstaben a, b oder c enthalten.

```
S,L'a',>L'a'+<L'z',L'z'
```
Es werden alle Sätze gesucht, die mindestens einen Kleinbuchstaben enthalten.

`S,'A'+'D1','B'+'D1','A'+'D2','B'+'D2'`
Es werden alle Datensätze gesucht, die die Zeichen A und D1 oder B und D1 oder A und D2 oder B und D2 enthalten. Die Suchfrage kann durch die algebraische Schreibweise `('A','B')+('D1','D2')` abgekürzt werden.

Mehrfachsuche - Format 2

S [-] [*n*] , ( ) [, A]

Durch () wird eine spezielle Such-Maske angefordert. Es stehen dem Benutzer damit 18 Bildschirmzeilen zur Eingabe umfangreicher Suchargumente zur Verfügung. Der Zusatz A hat die gleiche Bedeutung wie im Format 1 des Suche-Kommandos.

Die Zeilen 4 bis 21 der Suche-Maske dienen zur Aufnahme der Suchargumente. Diese sind lückenlos, d.h. ohne zusätzliche Blanks einzugeben.

Durch Betätigung der ENTER-Taste wird der Suchvorgang gestartet. Die K1-Taste bewirkt den Abbruch der Definition des Sucharguments. Mit der F3-Taste kann die ausgefüllte Suche-Maske in einer Hardcopy-Datei festgehalten werden.

Im Kommandofeld können BS2000-Kommandos, sowie das SP-Kommando eingegeben werden.

SP %*name* [,*jrsave*]

Save Parameters. Mit diesem Kommando wird die in der Suche-Maske eingetragene Suchbedingung unter dem angegebenen Namen in der Datei CFS.JRSAVE [.*jobname*] bzw. in einer mit /FILE *datei*,LINK=JRSAVE zugewiesenen Datei gesichert, sofern im SP-Kommando der Zusatz *jrsave* nicht angegeben wurde. Die genaue Regel, nach der der Name der JRSAVE-Datei gebildet wird, ist auf Seite **564** unter CFS.JRSAVE beschrieben.

%*name* — ein- bis 44-stelliger alphanumerischer Schlüsselbegriff für den abzuspeichernden Parameter-Set.

jrsave — Name der JRSAVE-Datei. Der Name sollte nur dann angegeben werden, wenn die Parameter in eine andere als die von CFS standardmäßig verwendete JRSAVE-Datei (siehe oben) gesichert werden.

Ein Beispiel für das Ausfüllen der Suche-Maske und für die Anwendung des Kommandos SP %*name* finden Sie im nächsten Abschnitt "Suchen mit Ersetzen".

Der Benutzer kann auf ein mit SP gesichertes Suchargument zurückgreifen durch das Kommando S,%*name* (siehe unten), bzw. S?,%*name* (Seite **310**).

Mehrfachsuche - Format 3

S [-] [*n*] , %*name* | %? [, *jrsave*] [, A]

Es werden alle Datensätze gesucht, die eine komplexe Suchbedingung erfüllen, die in der Suche-Maske mit dem Kommando SP %*name* in einem Parameter-Set gesichert wurde (siehe Seite **315**). Der optionale Zusatz A hat die gleiche Bedeutung wie beim oben beschriebenen Format 1 des Suche-Kommandos.

%? — Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Markieren mit 'x' kann einer der vorgeschlagenen Parametersets ausgewählt werden. Durch Markieren mit 'm' wird der Inhalt des Parametersets angezeigt und kann noch modifiziert werden.

Mehrfachsuche - Format 4

S [-] [*n*] , (*s-dat* | *bibl* ([*t*/] *elem*)) [, A]

s-dat | *bibl* ([*t*/] *elem*)

Name der Datei bzw. des Bibliothekselements, in dem die Suchitems gespeichert sind. Falls bei einem Bibliothekselement der Elementtyp nicht angegeben ist, wird S/ ergänzt, sofern es sich um eine PLAM-Bibliothek handelt.

Mit der Suche-Maske (siehe Seite **315**) können mehrfache Suchargumente mit maximal 15*80 Zeichen angegeben werden. Diese Beschränkung gilt nicht für die Suche-Datei *s-dat*. Für diese Datei besteht keine Einschränkung bezüglich der Größe und der Anzahl der Suchargumente.

Format der Datei *s-dat*:

Jeder Datensatz in *s-dat* beschreibt eine Suchbedingung, die mit der im nächsten Datensatz enthaltenen Suchbedingung verknüpft wird. Falls am Ende des Datensatzes keines der Verknüpfungszeichen **,/+/*** angegeben wurde, so wird standardmäßig die Oder-Bedingung als Verknüpfung mit dem Suchbegriff im nächsten Datensatz angenommen. Innerhalb eines Datensatzes können mehrere Suchitems mit dem Oder-, Und-, Wildcard-Zeichen verknüpft werden. Die maximale Länge von 250 Bytes sollte jedoch in keinem Datensatz überschritten werden.

A
Dieser optionale Zusatz hat die gleiche Bedeutung wie beim oben beschriebenen Format 1 des Suche-Kommandos.

Beispiel:

`S,(NAMEN)`
Es werden in der aktuellen Display-Datei alle Datensätze gesucht, die mindestens einen der in der Datei NAMEN aufgeführten Suchbegriffe enthalten. Ein Suchbegriff wird durch einen Satz in der unten stehenden Datei festgelegt.
Die Datei NAMEN habe folgenden Inhalt:

```
'ALBERT'
'ANDREAS'
'AMADEUS'+'THEODOR'+'ERNST'
........
'CARL'*'PHILIP'*'EMANUEL'
```

Und-Verknüpfung (+): Ein Satz mit dem String 'AMADEUS' wird nur dann als Treffer gewertet, wenn im gleichen Satz auch die Strings 'THEODOR' und 'ERNST' enthalten sind. Die Reihenfolge der einzelnen Items ist bei der Suche mit dem Verknüpfungszeichen '+' ohne Bedeutung. Ein Satz mit 'ERNST THEODOR AMADEUS' würde z.B. die Bedingung erfüllen.

Wildcard-Verknüpfung (*): Ein Satz mit dem String 'CARL' wird nur dann als Treffer gewertet, wenn im gleichen Satz an späterer Stelle die Strings 'PHILIP' und 'EMANUEL' enthalten sind.

Es ist möglich, die Suche jedes einzelnen Strings auf einen bestimmten Spaltenbereich zu begrenzen.

Im oben aufgeführten Beispiel könnten die Hochkommas vor und nach den Suchstrings weggelassen werden, da die gesuchten Zeichenfolgen keine Blanks oder andere Sonderzeichen enthalten.

Suchbegriffe, die Kleinbuchstaben enthalten, können in der Form L'...' angegeben werden. In diesem Fall wird die zwischen den Hochkommas stehende Zeichenfolge nicht in Großbuchstaben umgewandelt.

Für eine Beschreibung der Syntax und der verschiedenen Abkürzungsmöglichkeiten siehe Seite 309.

Suchen mit Ersetzen

Vorbemerkung: Das Kommando "Suchen mit Ersetzen" funktioniert in der gewünschten Art nur im Modify-Modus (Kommando M). Im Non Modify-Modus wird lediglich gezählt, wie oft eine Ersetzung stattgefunden hätte (Anzeige der Anzahl der Treffer).

S [-] [*n*], **...=** [*len*] *item2* [, ...= [*len*] *item2*] .... [, *opt*]

S.... steht für ein einfaches Suchargument wie im Abschnitt "Suchen von Zeichenfolgen (einfaches Suchargument)" beschrieben.

len — wahlweise Längenangabe. Beim Ersetzungsvorgang wird *item2* solange wiederholt, bis die angegebene Länge *len* erreicht ist. Falls nicht angegeben: *len* = Länge des Ersetzungsstrings.

item2 — Ersetzungsstring: C'...' | L'...' | X'...'

Über das Kommando SET REPLACEMODE=F|L kann bestimmt werden, welcher Suchstring bei einer Verknüpfung S,'*src1*'*'*src2*'='*str3*' durch *str3* ersetzt wird: Standardmäßig der erste (*src1*) oder der letzte (*src2*).

Das Konstrukt: ...= [*len*] *item2* kann beliebig oft wiederholt werden. Als Trennzeichen sind Kommas zu verwenden (Oder-Bedingungen).

Falls die Kommandozeile von CFS hierfür nicht ausreicht, kann durch S,()[,*opt*] eine eigene Suchmaske angefordert werden. Auch die Formate 3 und 4 des Suche-Kommandos können für das Suchen mit Ersetzen benützt werden:
S, %*name* [, *opt*] | S, (*s-dat*) [, *opt*].

Die Optionen [,*opt*] können auch in der Suchmaske/in der Datei *s-dat* im Anschluß an das letzte Paar *suchstring=ersatzstring* angegeben werden.

opt — C | R | A | Q
wahlweise Zusatzoptionen zur Steuerung des Ersetzungsvorgangs.

C — Die C-Option (Change) bewirkt einen echten Austausch von Suche- und Ersetzungsstring in den Datensätzen. Die Satzlängen werden in diesem Fall entsprechend angepaßt. Die Option C bewirkt eine Ersetzung analog der des EDT.

Standardmäßig (ohne C-Option) erfolgt die Ersetzung in der Weise, daß der gefundene Suchstring durch den angegebenen Ersatzstring überschrieben wird. Die Länge des modifizierten Datensatzes ändert sich auch dann nicht, wenn Suchstring und Ersetzungsstring verschiedene Länge haben.

Die C-Option ist nur bei ISAM-Dateien bzw. bei in ISAM-Dateien umgesetzten SAM-Dateien anwendbar. RECFORM=V ist ebenfalls Voraussetzung für die C-Option.

R — Reverse. Der Suche- und Ersetzungsvorgang innerhalb der Datensätze erfolgt in umgekehrter Reihenfolge von rechts nach links. In diesem Modus können keine mehrfachen Suche- und Ersetzungsstrings angegeben werden.

In Kombination mit der R-Option kann nur die C-Option (siehe oben) angegeben werden: CR.
Standard: Die Ersetzung erfolgt von links nach rechts.

A **Alle Treffer** innerhalb eines Datensatzes werden ersetzt (analog der EDT-Anweisung: @ON...**CA**'...'T'...').
Standard: Die Ersetzung wird nur beim ersten Treffer in einem Datensatz durchgeführt.

Falls weder die C-, noch die A-Option angegeben wurde, so wird der erste Treffer eines jeden Datensatzes durch den angegebenen Ersetzungsstring überschrieben.

Q Query: Vor jedem Ersetzungsvorgang wird der Benutzer um Bestätigung gebeten. Das Sichtfenster wird 20 Spalten links vom Trefferstring positioniert. Durch Betätigung der ENTER-Taste wird der Trefferstring durch den Ersatzstring ausgetauscht. Durch Betätigung der K3-Taste wird der Ersetzungsvorgang nicht ausgeführt. In beiden Fällen (ENTER/K3) wird nach Betätigung der Datenübertragungstaste auf den nächsten Treffer positioniert. Durch Überschreiben des im Kommandofeld vorgegebenen S-Kommandos mit Blank wird die Suche abgebrochen.

Standard: kein Query-Modus. Die Trefferstrings werden ohne Benutzeranfrage durch die Ersatzstrings ausgetauscht.

Hinweise:

Die C-, A-, und Q-Option können auch kombiniert werden: CA / CQ / AQ / CAQ. Die Reihenfolge der Kombination ist hierbei nicht frei wählbar. Zuerst muß die C-Option angegeben werden falls gewünscht, danach die A-Option und als letztes die Q-Option. CA entspricht dem Change All des EDT. CAQ ist eine Variante, die vor jeder Ersetzung eine Bestätigung durch den Benutzer verlangt.

Bei Anwendung der Option CAQ (Change All Query) und bei unterschiedlicher Länge des Such- und Ersetzungsstrings kann der Fall eintreten, daß am Ende des aktuellen Treffersatzes einige Zeichen nicht dargestellt werden (Satz erscheint abgeschnitten). Dieser Effekt hat keinerlei Auswirkungen auf die tatsächlichen Daten.

Falls beim Suchen mit Ersetzen die C-Option nicht angegeben wurde, so ist folgendes zu beachten:

Ist der Ersatzstring (*item2*) länger als der Suchstring (*item1*), so werden Daten, die rechts von *item1* stehen, durch *item2* überschrieben.

Ist der Ersatzstring (*item2*) kürzer als der Suchstring (*item1*), so wird der rechtsstehende Teil von *item1* nicht verändert.

ISAM: innerhalb des Schlüssels wird *item1* nicht gesucht und es findet demzufolge auch keine Ersetzung statt.

PAM: Suchen über Blockgrenzen
Bei PAM-Dateien muß der zu suchende String nicht vollständig in einem Datenblock enthalten sein. Falls der Suchbegriff (z.B. 'DATABASE') am Ende eines PAM-Blocks beginnt und sich am Anfang des nächsten PAM-Blocks fortsetzt (z.B. 'DA' + 'TABASE'), so wird diese Konstellation von CFS ebenfalls als Treffer erkannt und durch den Ersatzstring ausgetauscht.

Beispiele:

```
S,C'C''passw'''=C'C''passw-neu''',CAQ
```
Im ISAM-Modus wird nach der Zeichenfolge C'passw' gesucht. Bei jedem Treffer wird der Benutzer gefragt, ob der String durch C'passw-neu' ersetzt werden soll. ENTER-Taste: Ersetzung vornehmen und weitersuchen. K3-Taste: Keine Ersetzung und wei-

tersuchen. Durch das Austauschen der Strings findet eine Verschiebung der rechtsstehenden Daten statt.

`S,:9:''=60X'FF'`
Vom ersten angezeigten Satz bis Dateiende werden die Daten ab Spalte 9 (Satzanfang = Spalte 1) in der Länge 60 auf X'FF' gesetzt (überschrieben).

`S,:90:X'FF'+:1990:X'FA00'+:1545:''=5X'FF'`
Vom ersten angezeigten Satz bis Dateiende werden alle Sätze gesucht, die sowohl in Spalte 90 X'FF' als auch in Spalte 1990 die Zeichenfolge X'FA00' enthalten. Bei allen, diese Bedingung erfüllenden Datensätzen werden die Spalten 1545 bis 1549 mit X'FF' überschrieben.

`S,'ABCD'='',CA`
Vom ersten angezeigten Satz bis Dateiende wird die Zeichenfolge ABCD im gesamten Spaltenbereich gesucht und aus dem Satz entfernt. Das Entfernen der Zeichenfolge kann in einem Satz an mehreren Stellen erfolgen (Option A). Die Verarbeitungsoption C hat zur Folge, daß die resultierenden Sätze in ihrer Länge verkürzt werden.

`S,()`
Aufgrund dieses Kommandos wird die Suche-Maske ausgegeben, in die der Benutzer folgende Such- und Ersetzungsanweisung einträgt:

```
dd.mm.yy   hh:mm:ss           HOST: ........                    TSN: ....
COMMAND :SP %cap
please enter/modify search-item below
S,L'a'='A',L'b'='B',L'c'='C',L'd'='D',L'e'='E',L'f'='F',L'g'='G',L'h'='H',L'i'=
'I',L'j'='J',L'k'='K',L'L'='L',L'm'='M',L'n'='N',L'o'='O',L'p'='P',L'q'='Q',L'r
'='R',L's'='S',L't'='T',L'u'='U',L'v'='V',L'w'='W',L'x'='X',L'y'='Y',L'z'='Z',L
'ae'='Ae',L'oe'='Oe',L'ue'='Ue',L'ä'='Ä',L'ö'='Ö',L'ü'='Ü',A



ENTER: start search / K1: abort / Command  SP %name: save search-item / F3: HC
Activate / Redisplay saved search-item with Command: S,%name / S?,%name
param-set saved as %CAP      in CFS.JRSAVE.HILBERT
```

Vom ersten angezeigten Satz bis Dateiende werden alle Kleinbuchstaben (siehe Parameter A am Ende) in die entsprechenden Großbuchstaben umgewandelt. Durch das Kommando SP %CAP wird die Suche- und Ersetzungsanweisung unter dem Namen %CAP in der Datei CFS.JRSAVE [*.jobname*] bzw. in einer mit /FILE *datei*, LINK=JRSAVE zugewiesenen Datei gespeichert. Die genaue Regel, nach der der Name der JRSAVE-Datei gebildet wird, ist auf Seite **565** ("von CFS angelegte Dateien/Jobvariablen") beschrieben.

`S,%CAP`
Die unter dem Namen %CAP gespeicherte Suche- bzw. Ersetzungsanweisung wird zur Ausführung gebracht.

`S?,%CAP`
Die unter dem Namen %CAP gespeicherte Suche- bzw. Ersetzungsanweisung wird am Bildschirm angezeigt (Maske siehe oben) und kann danach zur Ausführung gebracht werden.

Suchen mit Löschen der gefundenen Sätze/Strings

S ...=E [I | P | PI | S | SI] [, Q] [, R]

S ... steht für eine einfache oder zusammengesetzte Suchanweisung wie in den vorhergehenden Abschnitten "Suchen von Zeichenfolgen (einfaches Suchargument / mehrere Suchargumente)" beschrieben.

Wird die Zeichenfolge in einem Datensatz gefunden, so werden folgende Löschoperationen ausgeführt:

E — Löschen des gesamten Datensatzes.

EI [,A] — Löschen des gefundenen Suchitems. Der Datensatz wird entsprechend verkürzt.
EI,A: Es werden alle Trefferitems im Satz gelöscht. In diesem Fall darf die Option ,R (Reverse) nicht angegeben werden.
Standard: Es wird das erste gefundene Suchitem im Satz gelöscht.

EP — Löschen vom Ende des ISAM-Schlüssels bis zum Beginn des Suchitems (Prefix).

EPI — Löschen vom Ende des ISAM-Schlüssels bis zum Ende des Suchitems (Prefix + Item).

ES — Löschen vom Ende des Suchitems bis zum Ende des Satzes (Suffix).

ESI — Löschen vom Anfang des Suchitems bis zum Ende des Satzes (Suffix + Item).

Q — Query: Der Benutzer wird bei jedem die Suchbedingung erfüllenden Treffer gefragt, ob die Löschoperation durchgeführt werden soll. Durch Betätigung der ENTER-Taste wird die vorgeschlagene Löschoperation bestätigt und zum nächsten Treffersatz positioniert. Durch Betätigung der K3-Taste wird die Löschoperation nicht ausgeführt und der nächste Treffer angeboten. Durch Überschreiben des im Kommandofeld stehenden S-Kommandos wird die Suche abgebrochen.

Standard: kein Query-Modus. Das Löschen wird ohne Benutzeranfrage ausgeführt.

R — Reverse. Der Suche- und Ersetzungsvorgang innerhalb der Datensätze erfolgt in umgekehrter Reihenfolge von rechts nach links. In diesem Modus können keine mehrfachen Suche- und Ersetzungsstrings angegeben werden.

Standard: Die Ersetzung erfolgt von links nach rechts.

Hinweis:

Die oben beschriebenen Varianten des Suche-Kommandos sind nur im Modify-Modus und nur für ISAM-Dateien zulässig. SAM-Dateien müssen zuvor mit dem Kommando IS[AM] in das ISAM-Format umgewandelt werden.

Beispiele:

```
S,:1:>'0100'+:1:<'0500'=E
```

Vorausgesetzt wird eine ISAM-Datei mit Recform=V,Keypos=5,Keylen=8. Vom ersten angezeigten Satz an werden alle Sätze mit ISAM-Schlüsseln größer als 0100 und kleiner als 0500 gelöscht.

`S,:10-12:'1',:10-12:'2',:10-12:'3'=E,Q`
Dem Benutzer wird jeder Datensatz, der in den Spalten 10 bis 12 die Zeichen 1, 2 oder 3 enthält zum Löschen angeboten (Query-Modus, Parameter Q). Durch Betätigung der ENTER-Taste wird der angezeigte Satz gelöscht und der nächste Treffersatz aufgesucht. Durch Betätigung der K3-Taste wird der angezeigte Satz nicht gelöscht und der nächste Treffersatz aufgesucht.

Suchen mit direkter Ausgabe der gefundenen Sätze am Bildschirm

S ...=P [L]

Wird die Zeichenfolge in einem Datensatz gefunden, so wird der Satzinhalt im Line-Modus am Bildschirm ausgegeben und die Suche fortgesetzt. Diese Form des Suche-Kommandos entspricht der Anweisung ON&P'....' des EDT.

S ... steht für eine einfache oder zusammengesetzte Suchanweisung wie in den vorhergehenden Abschnitten "Suchen von Zeichenfolgen (einfaches Suchargument / mehrere Suchargumente)" beschrieben.

Bei Dateien mit FCBTYPE=SAM und PAM wird dem Satzinhalt die entsprechende Satz- bzw. Blocknummer vorangestellt. Nach Ausgabe von maximal 21 Sätzen wird eine Überlaufkontrolle eingefügt ("Please acknowledge / End: K1"). Durch Betätigung der K1-Taste wird der Suchvorgang beendet. Jede andere Eingabe bewirkt eine Fortsetzung der Suche bis zum Dateiende bzw. bis zum Erreichen der im Suche-Kommando angegebenen maximalen Satzanzahl (S*nn*,'...'=P).

Nach Beendigung des Suchvorgangs wird der Display-Bildschirm wieder auf den gleichen Satzausschnitt positioniert wie vor Beginn des Suche-Kommandos.

Standardmäßig wird pro Treffersatz am Bildschirm nur eine Zeile (80 Byte) ausgegeben. Befindet sich das Suchitem nicht in den ersten 80 Bytes des Satzes, so wird ein Ausschnitt des Treffersatzes angezeigt, der 40 Bytes zur linken und rechten Seite des Suchitems darstellt.

PL

Print Long. Die Treffersätze werden bis zu einer maximalen Länge von 2048 Bytes am Bildschirm dargestellt.

Hinweis:

Nach Beendigung der Anzeige der Treffer ist die Suchbedingung noch gespeichert. Durch Eingabe des Kommandos S kann das Sichtfenster zum ersten Satz positioniert werden, der das Suchitem enthält.

Beispiele:

`S,'MVC'*'BLANK'=P`
Sucht vom ersten, im Sichtfenster angezeigten Satz bis Dateiende nach Sätzen, die die Zeichenfolge 'MVC' und rechts davon 'BLANK' enthalten. Die gefundenen Sätze werden im Line-Modus am Bildschirm ausgegeben. Nach Beendigung des Suchvorgangs wird das Sichtfenster auf den Satz positioniert, bei dem das Suche-Kommando aufgesetzt wurde.

`S100,:100-200:X'47'=PL`
Sucht in den nächsten 100 Sätzen/Datenblöcken jeweils im Spaltenbereich 100 - 200 nach der Zeichenfolge X'47'. Die gefundenen Sätze werden in der vollen Länge am Bildschirm ausgegeben.

Suchen mit Wegschreiben der Treffer

S ....=W [*datei* | LINK=WFCB [,C] [,E|O] [,Q] [,BLK] [, *f-attr*|?] [/[*:col:*] [*len*] [, NS]]]

S ... steht für eine einfache oder zusammengesetzte Suchanweisung wie in den vorhergehenden Abschnitten "Suchen von Zeichenfolgen (einfaches Suchargument / mehrere Suchargumente)" beschrieben.

datei Die Treffersätze werden in die angegebene Datei oder falls kein Dateiname wurde, in die zuletzt eröffnete Write-Datei geschrieben.

LINK=WFCB Die Treffersätze werden in die mit /FILE ...,LINK=WFCB,... zugewiesene Datei geschrieben.

C Die Ausgabedatei wird nach dem letzten geschriebenen Satz geschlossen.
E Die Ausgabedatei wird mit Open=Extend eröffnet.
O Eine bestehende Ausgabedatei wird überschrieben (Open=Output).
Standard: siehe Hinweise.

Q Query: Der Benutzer wird bei jedem die Suchbedingung erfüllenden Satz gefragt, ob dieser Satz in die Write-Datei geschrieben werden soll. Durch Betätigung der ENTER-Taste wird der Satz weggeschrieben und zum nächsten Treffersatz positioniert. Durch Betätigung der K3-Taste wird der Satz nicht weggeschrieben und der nächste Treffer angeboten. Durch Überschreiben des im Kommandofeld stehenden S-Kommandos wird die Suche abgebrochen.
Standard: kein Query-Modus. Die Treffersätze werden ohne Benutzeranfrage in die Write-Datei geschrieben.

Die E/O und die Q-Option können, durch ein Komma getrennt, auch kombiniert angegeben werden.

BLK Write Block: Diese Option ist nur von Bedeutung, falls die aktuelle Display-Datei eine PAM-Datei ist und der Pamkey-Modus eingeschaltet ist (Kommando PK). Aufgrund der Option BLK werden die zu den Pam-Keys gehörigen Blöcke in die Write-Datei übertragen. Standard: Im Pamkey-Modus werden nur die Pam-Keys in eine SAM-Datei geschrieben.

f-attr Dateiattribute der Write-Datei entsprechend der Syntax eines FILE- oder ADD-FILE-LINK Kommandos.

Beispiel (Parameter gemäß FILE-Kommando):
```
FCBTYPE=SAM,RECFORM=F,RECSIZE=80,BLKSIZE=(STD,16),SPACE=(120
,30),VOLUME=PVT001,DEVICE=D3480
```

Beispiel (Parameter gemäß ADD-FILE-LINK Kommando):
```
ACCESS=*SAM,REC-F=*FIX,REC-S=80,BUFFER-L=*STD(16)
```

? Es wird der SDF-Fragebogen des Kommandos /ADD-FILE-LINK (ADFL) ausgegeben. Die vorgegebenen Werte der ersten beiden Parameter (LINK-NAME=WFCB und FILE-NAME=...) dürfen nicht verändert oder gelöscht werden.

Standard: Für die Ausgabedatei werden alle Dateiattribute der Display-Datei außer LOGLEN, VALLEN, VALPROP, DSPACE übernommen.
Falls *f-attr* angegeben, so werden nur diese Attribute für die Ausgabedatei verwendet. Es werden keine Attribute der Display-Datei in die Ausgabedatei übernommen. *f-attr* kann nicht zusammen mit LINK=WFCB (s.o.) angegeben werden.

col Es wird nur der Teil der Treffersätze ab Spalte :col: weggeschrieben (:1: bezeichnet den Satzanfang). Standard: Es wird ab Satzanfang weggeschrieben (:1:).

len Es wird ab der angegebenen Spalte/ab Satzanfang in der Länge *len* weggeschrieben. Standard: Es wird in der maximalen Länge (bis zum Satzende) geschrieben.

NS No Skip. Datensätze, bei denen die Spalte :col: außerhalb des beschriebenen Bereichs liegt, werden bei der Ausgabe in die Write-Datei nicht unterdrückt. Es wird ein Datensatz mit der Länge 4 erzeugt. Sätze dieser Art enthalten ein Satzlängenfeld, jedoch keine Daten.
Standard: Datensätze, bei denen die Spalte :col: außerhalb des beschriebenen Bereichs liegt, werden bei der Ausgabe in die Write-Datei unterdrückt.

Hinweise:

Bei ISAM-Dateien können im DUPKEY-Modus (Kommando DUPK) Sätze mit gleichen Schlüsseln in die Write-Datei übertragen werden. Bei ausgeschaltetem DUPKEY-Modus (Standard) wird bei mehreren Sätzen mit gleichen Schlüsseln nur der jeweils letzte in der Write-Datei gespeichert. Falls die Speicherung des jeweils ersten Datensatzes mit gleichen Schlüsseln gewünscht ist, so ist dazu die Suche vom Dateiende in Richtung Dateianfang auszuführen: S-,...=W...

Für die Write-Datei ist der DUPKEY-Modus wirksam, der zum Zeitpunkt des Eröffnens der Write-Datei (erstes W- / S..=W-Kommando) bestand. Ein nachträgliches Umschalten des DUPKEY-Modus hat auf die Write-Datei keine Wirkung, es sei denn, der Display-Modus wird vorübergehend beendet (Drücken der K1-Taste) und danach wieder aufgenommen.

Falls hinter dem Namen der Write-Datei keine der Optionen E/O (Extend/ Overwrite) angegeben wurde, so gilt folgende Regelung:

Falls die Write-Datei im aktuellen CFS-Lauf zum ersten Mal angesprochen wird, so wird als Open-Modus in jedem Fall O (Overwrite) angenommen, d.h. die Datei wird neu angelegt bzw. überschrieben.

Falls in mehreren W / S...=W-Kommandos nacheinander die gleiche Write-Datei angegeben wurde, so wird sie standardmäßig mit Open=Extend eröffnet.

Durch die E-/O-Option kann vom Benutzer ein vom Standardfall abweichender Open-Modus angegeben werden.

Beispiele:

```
S,:1:>X'0054'=W CFS.REC80
```
Zur Ermittlung der Datensätze mit einer bestimmten Länge kann mit dem Kommando O-4 das Satzlängenfeld in das Display miteinbezogen werden. Durch das angegebene Suche-Kommando (Syntax siehe Seite 309) werden in einer RECFORM=V Datei alle Datensätze gefunden, deren Datenbereich eine Länge von mehr als 80 Bytes aufweist. Die gefundenen Sätze werden in die Datei CFS.REC80 kopiert.

```
S,L'a',>L'a'+<L'z',L'z'=W CFS.KLEIN
```
Es werden alle Datensätze, die Kleinbuchstaben enthalten, in die Datei CFS.KLEIN geschrieben. Syntax siehe Seite 313.

```
S,'ABC'=W SAVE.ABC,FCBTYPE=SAM/:9:100
```
Vom ersten angezeigten Satz bis Dateiende wird in der Display-Datei der String 'ABC' gesucht. Jeder Satz, in dem mindestens ein Treffer gefunden wurde, wird ab Spalte 9 in der Länge 100 in die Ausgabedatei SAVE.ABC geschrieben. Die Datei SAVE.ABC soll unabhängig vom Format der im Display angezeigten Datei mit FCBTYPE=SAM angelegt werden. Standardmäßig, d.h. falls keine Dateiparameter angegeben wurden, erhält die Write-Datei die gleichen Datei- und Satzformate wie die Display-Datei.

Suchen mit Wegschreiben in eine CFS-Prozedurvariable

S ....=W, (&*var*) [/[*:col:*] [*len*]]

Der Datensatz der Display-Datei, der beim Suchvorgang einen Treffer gebracht hat, wird vollständig oder als Teil in die CFS-Prozedurvariable mit dem angegebenen Namen übertragen.

:col: Es wird nur der Teil des Displaysatzes ab Spalte :col: in die Prozedurvariable übertragen (:1: bezeichnet den Satzanfang).
Standard: Es wird der vollständige Datensatz übertragen.

len Es wird ab der angegebenen Spalte/ab Satzanfang in der in *len* angegebenen Länge geschrieben.
Standard: Es wird in der vollen Länge, d.h. bis zum Satzende weggeschrieben. Die Maximallänge ist jedoch in jedem Fall auf 80 Bytes begrenzt.

Hinweise:

Falls mehrere Datensätze im Verlauf der Suche als Treffer gemeldet werden, so wird der jeweils letzte in die CFS-Prozedurvariable übertragen. Bei der Suche vom Dateiende zum Dateianfang (S-) wird der erste, in der Datei enthaltene Treffer, in die Prozedurvariable geschrieben.

Bei ISAM-Dateien können im DUPKEY-Modus (Kommando DUPK) Sätze mit gleichen ISAM-Schlüsseln in die Write-Datei übertragen werden.
Bei ausgeschaltetem DUPKEY-Modus (NDUPK, Standard) wird bei mehreren Sätzen mit gleichen Schlüsseln nur der jeweils letzte in der Write-Datei gespeichert.

Wegschreiben von Sätzen aus Display-Datei

W[*n*][, *datei* | LINK=WFCB [, C|E|O|BLK]] [,*f-attr*|?] [/[*:col:*] [*len*] [, NS]] [, SC]

Vom ersten im Sichtfenster angezeigten Satz ausgehend werden *n* Sätze in die angegebene Write-Datei weggeschrieben. Das Sichtfenster wird um diese *n* Sätze weiter positioniert.

n Anzahl der in die Write-Datei zu übertragenden Sätze.
Das Zeichen '$' anstelle einer Anzahl *n* steht für die maximale Anzahl *n* = 99999.
Standard: *n* = 1.

datei Name der Datei, in die die Sätze zu schreiben sind (Write-Datei).
Falls weggelassen: Es wird in die zuletzt angegebene Write-Datei geschrieben.

LINK=WFCB Die Datensätze werden in die mit /FILE ...,LINK=WFCB,... zugewiesene Datei geschrieben.

C Die Ausgabedatei wird nach dem letzten geschriebenen Satz geschlossen.

E Die Ausgabedatei wird mit Open=Extend eröffnet.

O Eine bestehende Ausgabedatei wird überschrieben (Open=Output).

BLK Diese Option ist nur von Bedeutung, falls die aktuelle Display-Datei eine PAM-Datei ist und der Pamkey-Modus eingeschaltet ist (Kommando PK). Durch die Option BLK werden die zu den Pam-Keys gehörigen Blöcke in die Write-Datei übertragen. Standard: Es werden nur die Pam-Keys in die Datei geschrieben.

f-attr Dateiattribute der Write-Datei entsprechend der Syntax eines FILE- oder ADD-FILE-LINK Kommandos.

Beispiel (Parameter gemäß FILE-Kommando):
```
FCBTYPE=SAM,RECFORM=F,RECSIZE=80,BLKSIZE=(STD,16),SPACE=(120
,30),VOLUME=PVT001,DEVICE=D3480
```

Beispiel (Parameter gemäß ADD-FILE-LINK Kommando):
```
ACCESS=*SAM,REC-F=*FIX,REC-S=80,BUFFER-L=*STD(16)
```

? Es wird der SDF-Fragebogen des Kommandos /ADD-FILE-LINK (ADFL) ausgegeben. Die vorgegebenen Werte der ersten beiden Parameter (LINK-NAME=WFCB und FILE-NAME=...) dürfen nicht verändert oder gelöscht werden.

Standard: Für die Ausgabedatei werden alle Dateiattribute der Display-Datei außer LOGLEN, VALLEN, VALPROP, DSPACE übernommen.
Falls *f-attr* angegeben, so werden nur diese Attribute für die Ausgabedatei verwendet. Es werden keine Attribute der Display-Datei in die Ausgabedatei übernommen. *f-attr* kann nicht zusammen mit LINK=WFCB (siehe oben) angegeben werden.

:col: Es wird nur der Teil der Displaysätze ab Spalte :col: weggeschrieben (:1: = Satzanfang).
Standard: Es werden die vollständigen Datensätze übertragen (:1:).

len Es wird ab der angegebenen Spalte/ab Satzanfang in der in *len* angegebenen Länge geschrieben.
Standard: Es wird in der vollen Länge, d.h. bis zum Satzende weggeschrieben.

NS No Skip. Datensätze, bei denen die Spalte :col: außerhalb des beschriebenen Bereichs liegt, werden bei der Ausgabe in die Write-Datei nicht unterdrückt. Es wird ein Datensatz mit der Länge 4 erzeugt. Sätze dieser Art enthalten ein Satzlängenfeld, jedoch keine Daten.
Standard: Datensätze, bei denen die Spalte :col: außerhalb des beschriebenen Bereichs liegt, werden bei der Ausgabe in die Write-Datei unterdrückt.

SC Ist im Display ein Zeilenlineal eingeblendet (Scale), so wird der Inhalt dieses Zeilenlineals als erster Satz in die Write-Datei geschrieben.
Standard: Das Zeilenlineal wird nicht in die Ausgabedatei aufgenommen.

Hinweise:

Die Write-Datei wird beim Verlassen des Display-Modus über die K1-Taste bzw. Kommando LST automatisch geschlossen. Bei einem nachfolgenden W-Kommando ohne Angabe eines Dateinamens (W [*n*]) wird die zuletzt benutzte Write-Datei im EXTEND-Modus (SAM-Datei) bzw. im INOUT-Modus (ISAM-/PAM-Datei) eröffnet.

Wurde hinter dem Namen der Write-Datei keine der Optionen E/O (Extend/ Overwrite) angegeben, so gilt folgende Regelung:

Falls die Write-Datei im aktuellen CFS-Lauf zum ersten Mal angesprochen wird, so wird als Open-Modus in jedem Fall O (Overwrite) angenommen, d.h. die Datei wird neu angelegt bzw. überschrieben.

Falls in mehreren W / S...=W-Kommandos nacheinander die gleiche Write-Datei angegeben wurde, so wird sie standardmäßig mit Open=Extend eröffnet.

Durch die E-/O-Option kann vom Benutzer ein vom Standardfall abweichender Open-Modus angegeben werden.

Bei ISAM-Dateien können im DUPKEY-Modus (Kommando DUPK) Sätze mit gleichen ISAM-Schlüsseln in die Write-Datei übertragen werden.
Bei ausgeschaltetem DUPKEY-Modus (NDUPK, Standard) wird bei mehreren Sätzen mit gleichen Schlüsseln nur der jeweils letzte in der Write-Datei gespeichert.

Für die Write-Datei ist der DUPKEY-Modus wirksam, der zum Zeitpunkt des Eröffnens der Write-Datei (erstes Write-Kommando) bestand. Ein nachträgliches Umschalten des DUPKEY-Modus hat auf die Write-Datei keine Wirkung, es sei denn, der Display-Modus wird vorübergehend beendet (Drücken der K1-Taste) und danach wieder aufgenommen.

Mit dem Write-Kommando kann eine sequentielle Datei nach einem Schlüsselbegriff aufsteigend sortiert werden:

`W9999,datei2,FCBTYPE=ISAM,KEYPOS=kk,KEYLEN=ll`

kk und ll bezeichnen den in den Datensätzen der sequentiellen Datei enthaltenen Schlüsselbegriff. Falls mehrere Datensätze mit gleichem Schlüsselbegriff existieren, muß vor Write das Kommando DUPK (siehe oben) eingegeben werden. Die nach dem gewünschten Kriterium sortierte ISAM-Datei kann anschließend wieder in eine sequentielle Datei zurückkonvertiert werden. Die Konvertierung einer SAM-Datei in eine ISAM-Datei mit gleichzeitiger Umsortierung gemäß einem vorgegebenen Schlüsselbegriff kann auch über die Variable Action ONXCONV vorgenommen werden.

Das Write-Kommando ist prinzipiell auch dazu geeignet, mehrere Dateien/Bibliothekselemente in einer Ausgabedatei zusammenzufügen.
Ein einfacheres Mittel hierfür bietet die Variable Action ONXLIST... unter Angabe der NH-Option (No Header). Mehr hierzu auf Seite **Fehler! Textmarke nicht definiert.**.

Beispiel:

Die im folgenden aufgeführte CFS-Prozedur sucht in der aktuellen Display-Datei alle Sätze mit einem vorgegebenen String und schreibt *n* Sätze davor und *m* Sätze danach in die Datei &W-FILE.

```
*PROC N,(&ITEM,&W-FILE,&M,&N)
SR
*&ANZ=&M+&N
*&ANZ=&ANZ+1
S,&ITEM
*IF &CFSMSG(1,20) -= '>> SYNTACTICAL ERROR'  SKIP .LOOP
*WRITESYS '&CFSMSG'
*SKIP .ENDE2
.LOOP
*IF &CFSMSG(53,3) = 'NOT'  SKIP .ENDE
-&M
W&ANZ,&W-FILE
S
*SKIP .LOOP
.ENDE
*K1
.ENDE2
```

Wegschreiben eines Display-Satzes in eine CFS-Prozedurvariable

W,(&*var*) [/[*:col:*] [*len*]]

Der erste im Sichtfenster angezeigte Datensatz der Display-Datei wird vollständig oder als Teil in die CFS-Prozedurvariable mit dem angegebenen Namen übertragen.

:col: Es wird nur der Teil des Displaysatzes ab Spalte :col: in die Prozedurvariable übertragen (:1: = Satzanfang).
Standard: Es wird der vollständige Datensätze übertragen.

len Es wird ab der angegebenen Spalte/ab Satzanfang in der in *len* angegebenen Länge geschrieben.
Standard: Es wird in der vollen Länge, d.h. bis zum Satzende weggeschrieben. Die maximale Länge ist jedoch auf 80 Bytes begrenzt.

Editieren von ISAM-Dateien beliebigen Formats

Im Modify-Modus können ISAM-Dateien mit erweiterten und zum Teil EDT-ähnlichen Funktionen bearbeitet werden. Datensätze können z.B. kopiert, gelöscht, zusammengefügt und in ihrer Position innerhalb der Datei verschoben werden. Auch das Einfügen und Löschen von Spaltenbereichen, sowie das Aufsplitten von Sätzen wird unterstützt.

Sätze kopieren/löschen/zusammenfügen/verschieben

C | R | M **Datensätze in den Kopierpuffer eintragen**. Erste Stelle des ISAM-Schlüssels der zu kopierenden Sätze mit dem Zeichen C (Copy), R (Retain) oder M (Move) überschreiben. Die restlichen Stellen der Schlüssel dürfen nicht verändert werden. In der letzten Bildschirmzeile wird die Gesamtanzahl der momentan im Puffer enthaltenen Sätze angezeigt. Beim Eintragen von Sätzen in den Kopierpuffer mit C/M ist der Puffer, wie im EDT, nach jedem Einfügen des Kopierpuffers mit B (Before) oder A (After) gelöscht. Die Eintragung von Sätzen in den Kopierpuffer mit R (Retain) bewirkt, daß dieser nach dem Einfügen mit B oder A nicht gelöscht wird und für weitere Einfügeoperationen zur Verfügung steht. Ein nicht gelöschter Kopierpuffer steht auch nach dem Wechsel der zu bearbeitenden Datei noch zur Verfügung.

***** **Kopierpuffer zurücksetzen**. Durch diese Markierung in der ersten Stelle eines ISAM-Schlüssels wird der Kopierpuffer zurückgesetzt (gelöscht), ohne daß ein Kopiervorgang vorgenommen wurde.

A | B **Kopierpuffer nach/vor einem Datensatz einfügen**. Erste Stelle des ISAM-Schlüssels mit dem Zeichen B (Before) oder A (After) überschreiben. Die restlichen Stellen des Schlüssels dürfen nicht verändert werden. Es können aus dem Kopierpuffer nur soviele Sätze eingefügt werden, wie die Differenz der Schlüssel vor und nach der Einfügeposition zuläßt. Z.B können zwischen zwei Sätzen mit den Schlüsseln 00011100 und 00011200 maximal 99 Sätze eingefügt werden. Wurden die Sätze mit der Markierung M (Move) in den Kopierpuffer eingetragen, so wird für jeden eingefügten Satz der entsprechende Originalsatz gelöscht.

I[*n*][,*len*] **Einfügen** von Leersätzen. Vor dem markierten Satz werden *n*, aus Blanks bestehende Leersätze eingefügt. Die einstellige Anzahl *n* muß direkt nach dem Markierungszeichen I angegeben werden. Falls kein Wert für *n* angegeben wurde, werden 9 Sätze eingefügt. Die optionale Angabe *len* spezifiziert die Länge der einzufügenden Sätze ohne ISAM-Schlüssel. *len* muß durch ein Blank vom restlichen Teil des ISAM-Schlüssels getrennt werden.
Falls kein Wert für *len* angegeben ist, wird als Standardwert *len*=72 verwendet.

J **Zusammenfügen** von Sätzen. Erste Stelle des ISAM-Schlüssels mit dem Zeichen J (Join) überschreiben. Der so markierte Satz wird an das Ende des vorhergehenden Satzes angehängt. Der ISAM-Schlüssel des mit J gekennzeichneten Satzes wird dabei ausgeblendet. Bei der Markierung mehrerer aufeinanderfolgender Sätze mit J werden diese gemeinsam an den vorhergehenden Satz angefügt.

E Sätze **löschen**. Erste Stelle des ISAM-Schlüssels mit E (Eliminate) überschreiben.

C Direktes **Kopieren** von Sätzen über ISAM-Schlüssel. Erste Stelle des ISAM-Schlüssels mit dem Zeichen C (Copy) überschreiben und an den restlichen Stellen die Position, an der der neue Satz eingefügt werden soll, eintragen. Für die Position des zu kopierenden Satzes wird die erste Stelle des ISAM-Schlüssels unverändert aus dem Ursprungssatz übernommen. Die Daten des zu kopierenden Satzes dürfen nicht zugleich mit dem Schlüssel verändert werden.

Direktes Verschieben von Sätzen über ISAM-Schlüssel. Werden ISAM-Schlüssel modifiziert, so verschieben sich die Datensätze in ihrer Position innerhalb der Datei entsprechend (Move).

Hinweise:

Der Kopierpuffer von CFS wird mit dem Verlassen des Display-Modus nicht automatisch gelöscht. Es ist daher möglich, mit den Markierungen C/R Datensätze aus einer Datei in eine/mehrere andere Dateien zu übertragen.

Werden die Daten im Hexadezimalmodus angezeigt, so sind anstelle der oben beschriebenen Markierungszeichen C/A/B/... die entsprechenden Hexadezimalwerte (X'C3'/X'C1'/X'C2') an der ersten Stelle der ISAM-Schlüssel einzutragen. Es besteht jedoch auch die Möglichkeit, die Markierungszeichen wie gewohnt im Character-Format einzugeben. Dazu ist das Zeichen im ersten Halbbyte des Schlüssels und ein Leerzeichen im zweiten Halbbyte des Schlüssels einzugeben.

Datensätze, deren ISAM-Schlüssel an der ersten Stelle bereits eines der Zeichen C/A/B/... aufweisen, können in der oben beschriebenen Weise nicht bearbeitet werden. Für diese Sonderfälle ist das Kommando RD anzuwenden (siehe unten).

Bei Move- und Copy-Operationen mit Modifikation des Schlüssels dürfen die Zielschlüssel nicht mit den Schlüsseln von bereits existierenden Sätzen übereinstimmen (DUPEKY=NO). Mit dem Kommando DUPK wird der Dupkey-Modus eingeschaltet. In diesem Fall sind doppelte ISAM-Schlüssel bei Move und Copy-Operationen möglich.

Weder im DUPK noch im NDUPK-Modus ist es möglich, aus einer Folge von Datensätzen mit gleichen ISAM-Schlüsseln gezielt den zweiten, dritten Satz, usw. mit Move/Copy/Eliminate zu bearbeiten. Bei Sätzen mit gleichen ISAM-Schlüsseln wird in jedem Fall der erste Satz mit diesem Schlüssel der entsprechenden Operation unterworfen.

Zeichen für Satzmarkierung redefinieren

RD *keyword=x* ReDefine.

Mit dem Kommando RD können die Markierungssymbole zur Bearbeitung von ISAM-Schlüsseln geändert werden.

keyword AFTER | BEFORE | CLEAR | COPY | ELIM | INSERT | JOIN | MOVE | RETAIN
Die Schlüsselwörter bestimmen das zu ändernde Markierungssymbol.

x neuer Wert des Markierungssymbols. Es kann jedes über die Tastatur eingebbare Zeichen gewählt werden. Standardmäßig sind folgende Werte definiert: AFTER=A, BEFORE=B, CLEAR=*, COPY=C, ELIM=E, JOIN=J, INSERT=I, MOVE=M, RETAIN=R.

Die Anzeige der aktuellen Markierungssymbole und deren Änderung ist auch über die SET-Maske (Kommando SET) möglich.

Beispiel:

`RD ELIM=D`
Durch das obige Kommando wird das Zeichen zum Löschen von Sätzen von 'E' in 'D' (=Delete) geändert.

Delete Characters (Spalten löschen)

DEL [*n*], [*:col:*] [*len*|*char*]

Beginnend mit dem ersten angezeigten Satz wird in *n* Sätzen der angegebene Spaltenbereich gelöscht. Die Länge der Sätze verkürzt sich entsprechend.

n Satzanzahl: 1 <= *n* <= 99999.
Das Zeichen '$' anstelle einer Anzahl *n* steht für die maximale Anzahl *n* = 99999.
Standard: *n* =1. Es wird nur im ersten angezeigten Satz gelöscht.

:col: 1 <= *col* <= 32000 | *col* = $ (Satzende).
erste zu löschende Spalte. Bei DEL..,:$:*len* werden vom jeweiligen Satzende nach links die Zeichen in der angegebenen Länge gelöscht.
Standard: Es wird der Bereich unmittelbar rechts vom ISAM-Schlüssel gelöscht (bei Dateien mit KEYPOS=5 und KEYLEN=8 ist dies die Spalte :9:).

len Länge des zu löschenden Spaltenbereichs ab der angegebenen Anfangs-Spalte *col* bzw. vom Satzende nach links (:$:).
Standard: *len* = 1.

char C'*x*'|X'*xx*'. Das angegebene Zeichen wird in jedem Satz von rechts nach links bis zum ersten Auftreten eines anderen Zeichens gelöscht. Falls das Zeichen *n* mal angegeben wird, so werden die letzten *n*-1 Zeichen am Satzende nicht gelöscht.
Diese Angabe ist nur sinnvoll in Verbindung mit der Spaltenangabe :$:.

Beispiele:

`DEL 9999,:500:10`
Vom ersten angezeigten Satz an werden in 9999 Sätzen (bzw. bis Dateiende) ab der Spalte 500 10 Zeichen gelöscht.

`DEL$,:$:10`
In allen Sätzen der Datei werden die letzten 10 Stellen gelöscht.

`DEL$,:$:C' '`
In den Sätzen der Datei werden Blanks am Satzende gelöscht.

`DEL$,:$:C'  '`
In den Sätzen der Datei werden alle Blanks am Satzende bis auf eines gelöscht.

Sätze löschen

ER [*n*] Eliminate Records. Beginnend mit dem zweiten am Bildschirm angezeigten Satz werden *n* Sätze gelöscht.

n Anzahl der zu löschenden Datensätze.
Das Zeichen '$' anstelle einer Anzahl *n* steht für die maximale Anzahl *n* = 99999.
Standard: *n* = 1.

ER *key1-key2* Beginnend mit *key1* werden alle Sätze mit Schlüsseln kleiner oder gleich bis *key2* gelöscht. Die Schlüssel können in einer der folgenden Darstellungsweisen angegeben werden: *string* | [C] '*string*' | X'*string*'. Die Schlüssel *key1* und *key2* müssen nicht vollständig, d.h. in der ganzen Länge, angegeben werden. Es genügt, den eindeutigen Anfang des gewünschten Anfangs- und Endeschlüssels zu spezifizieren. *key1* wird in diesem Fall rechts mit Low Value (X'00') und *key2* mit High Value (X'FF') aufgefüllt.

Beispiele:

`ER`
Der zweite am Bildschirm angezeigte Satz wird gelöscht.

`ER 999`
Beginnend mit dem zweiten am Bildschirm angezeigten Satz werden 999 folgende Sätze gelöscht.

`ER 0003-01`
Es werden alle Datensätze gelöscht, deren Schlüssel im Bereich zwischen '0003*' und '01*' liegen. * steht hier für eine Zeichenfolge beliebigen Inhalts, die sich bis zum Ende des Schlüssels erstreckt.

`ER X'AF'-X'AF'`
Es werden alle Datensätze gelöscht, deren Schlüssel mit X'AF' beginnen.

Leersätze einfügen

IL [*n*[+|-]], [*len*] [*item*]
Insert blank Lines. Zwischen dem ersten und dem zweiten angezeigten Satz werden *n* Leersätze eingefügt.

n Anzahl der einzufügenden Sätze. Standard: *n* = 1.
Die Schrittweite der eingefügten ISAM-Schlüssel berechnet sich aus der Differenz zwischen dem zweiten und dem ersten angezeigten Schlüssel dividiert durch die Anzahl *n*.

n+ Die Sätze werden mit der kleinstmöglichen Schrittweite beginnend vom unteren Schlüssel an eingefügt.

n- Die Sätze werden mit der kleinstmöglichen Schrittweite unterhalb des oberen Schlüssels eingefügt.

len Länge des Datenteils der einzufügenden Sätze. Der ISAM-Schlüssel und das Satzlängenfeld zählen bei der Länge des Datenteils nicht mit.
Standard: 72. Es werden Sätze mit 72 Datenbytes eingefügt.

item Zeichenfolge mit der die einzufügenden Sätze aufgefüllt werden sollen.
[C]'*string*' | L'*string*' | X'*string*'
Standard: C'␣'. Die einzufügenden Sätze enthalten Spaces.

Beispiele:

```
IL 100,256C'0123456789'
```

Zwischen dem ersten und zweiten angezeigten Satz werden 100 Sätze eingefügt, soweit es die Differenz der Schlüssel erlaubt. Die Länge der Sätze ergibt sich aus Länge des ISAM-Schlüssels + 4 Byte Satzlängenfeld + 256 Byte Daten. Die Daten der eingefügten Sätze werden auf 256 Stellen mit dem Muster C'0123456789' aufgefüllt. Der angegebene String wird im Ausgabesatz 25-mal voll und einmal in den ersten 6 Byte aneinandergereiht.

```
IL ,80
```

Nach dem ersten angezeigten Satz wird ein Satz mit 80 Blanks eingefügt.

Insert Characters (Spalten einfügen)

INS [*n*], [*:col:*] [*len*] [*item*]

Beginnend mit dem ersten im Sichtfenster angezeigten Satz wird in *n* Sätzen an der Spalte :col: in der Länge *len* die Zeichenfolge *item* eingefügt. Die Datensätze verlängern sich entsprechend.

n — Satzanzahl: 1 <= *n* <= 99999.
Das Zeichen '$' anstelle einer Anzahl *n* steht für die maximale Anzahl *n* = 99999.
Standard: *n* = 1. Insert wird nur im ersten angezeigten Satz durchgeführt.

:col: — 1 <= *col* <= 32000 | *col* = $ (Satzende).
Spalte, nach der eingefügt werden soll. Aufgrund von INS..., :$: ... erfolgt die Einfügung jeweils am Ende des Datensatzes.
Standard: Die Einfügung erfolgt unmittelbar nach dem ISAM-Schlüssel. Bei Dateien mit KEYPOS=5 und KEYLEN=8 ist dies die Spalte :9:

len — Länge der Einfügung. Standard: *len* = Länge des Items bzw. *len* = 1, falls *item* nicht angegeben wurde.

item — Einzufügende Zeichenfolge: C'*string*' | X'*string*'. C'*string*' kann auch als '*string*' angegeben werden. Falls die Länge von *item* kleiner ist als die angegebene Einfügelänge *len*, so wird *item* solange wiederholt, bis die in *len* angegebene Länge erreicht ist. Ist die Länge von *item* größer als *len*, so werden nur die ersten Stellen von *item* zur Einfügung benutzt.
Standard: C'␣', d.h. anstelle eines vorgegebenen Items werden Blanks eingefügt.

Beispiele:

```
INS 9999,:500:10X'FF'
```

Vom ersten angezeigten Satz an wird in 9999 Sätzen (bzw. bis Dateiende) ab Spalte 500 die Zeichenfolge X'FFFFFFFFFFFFFFFFFFFF' (=10X'FF') eingefügt.

```
INS :$:'1234567890'
```

Im ersten angezeigten Satz wird am Satzende die Zeichenfolge C'1234567890' angefügt.

```
INS :20:10
```

Im ersten angezeigten Satz werden ab der Spalte 20 10 Spaces eingefügt.

Sätze aus zweiter Datei einfügen

IR [*n*[+|-]] [, *datei*] [, *params*]

Es werden *n* Sätze aus der angegebenen Referenzdatei in die aktuelle Display-Datei kopiert.

n Anzahl der zu kopierenden Sätze.
Das Zeichen '$' anstelle einer Anzahl *n* steht für die maximale Anzahl *n* = 99999.
Standard: *n* = 1.
Die Schrittweite der eingefügten ISAM-Schlüssel berechnet sich aus der Differenz zwischen dem zweiten und dem ersten angezeigten Schlüssel dividiert durch die Anzahl *n*.

n+ Die Sätze werden mit der kleinstmöglichen Schrittweite beginnend vom unteren Schlüssel an eingefügt.

n- Die Sätze werden mit der kleinstmöglichen Schrittweite unterhalb des oberen Schlüssels eingefügt.

datei Referenzdatei, aus der die Sätze kopiert werden. Die Referenzdatei kann vom Fcbtyp SAM oder ISAM sein.
Standard: Falls *datei* nicht angegeben wurde, so wird auf die zuletzt eröffnete Referenzdatei zugegriffen.

params W | NW | K | KN | NK | B | C
oder eine Kombination dieser Werte getrennt durch Kommas.

W Wrap. Nach Erreichen des Endes der Referenzdatei wird der interne Zeiger wieder auf den Dateianfang gesetzt. Auf diese Weise können Datensätze aus der Referenzdatei wiederholt in die Display-Datei eingefügt werden.

NW No Wrap. Nach Erreichen des Endes der Referenzdatei bleibt der interne Zeiger dort stehen. Es wird kein weiterer Satz in die Display-Datei kopiert. Durch die B-Option (siehe unten) kann der interne Zeiger wieder auf den Beginn der Referenzdatei gesetzt werden.
Standard: NW.

K Keys. CFS fügt die Sätze der Referenzdatei entsprechend ihrem Originalschlüssel positionsgerecht in die Display-Datei ein (gestreute Einfügung).

Falls die Referenzdatei vom FCBTYPE=SAM ist, müssen die Daten für die ISAM-Schlüssel in den einzufügenden Sätzen enthalten sein.

Im K-Modus überschreiben Datensätze aus der Referenzdatei entsprechende Sätze mit den gleichen ISAM-Schlüsseln in der Display-Datei.
Im K-Modus ist ein zuvor abgesetztes DUPK-Kommando insofern wirksam, als Sätze aus der Referenzdatei mit gleichen ISAM-Schlüsseln wie in der Display-Datei, dort eingefügt werden können (kein Überschreiben). Dadurch ist es möglich, in der Display-Datei Datensätze mit gleichen ISAM-Schlüsseln zu erzeugen.

KN Keys No overwrite. Gleiche Funktion wie bei K (siehe oben), jedoch überschreiben Datensätze aus der Referenzdatei nicht Sätze mit den gleichen ISAM-Schlüsseln in der Display-Datei.

NK No Keys. Die in der Referenzdatei vorhandenen ISAM-Schlüssel werden beim Einfügen nicht mit übernommen. Die Referenzdatei wird wie eine sequentielle Datei behandelt. Die Sätze der Referenzdatei werden zwischen dem ersten und zweiten am Bildschirm angezeigten Satz in die Display-Datei eingefügt. Es werden neue ISAM-Schlüssel von CFS vergeben.
Standard: NK.

B Beginning. Die Referenzdatei wird wieder vom Anfang an gelesen. Diese Option sollte bei Ausgabe der Fehlermeldung "EOF on source-file" verwendet werden.

C Close. Die Referenzdatei wird nach dem Einfügen geschlossen. Standardmäßig bleibt die Referenzdatei solange geöffnet, bis CFS beendet oder ein IR-Kommando mit einer anderen Referenzdatei abgesetzt wird.

Hinweise:

Das IR-Kommando ist das Gegenstück zum W-Kommando bzw. zum Suche-Kommando mit kombiniertem Write (S...=W...). Mit dem Write-Kommando können bestimmte Sätze in eine Ausgabedatei von gleichem Format wie die Display-Datei geschrieben werden. Nach gesonderter Bearbeitung dieser Sätze in der Write-Datei können sie mit dem Kommando IR wieder in die Originaldatei eingespielt werden.

Falls sowohl die Display-, als auch die IR-Datei ISAM-Dateien sind, so müssen KEYPOS und KEYLEN in beiden Dateien übereinstimmen.

Bei Erreichen des Endes der Referenzdatei bleibt der interne Zeiger dort stehen. Durch IR [*n*],,B wird die Referenzdatei wieder vom Anfang gelesen.

Die Referenzdatei wird mit Verlassen des Display-Modus nicht geschlossen. Ein explizites Schließen der Referenzdatei kann mit der Option C im IR-Kommando (siehe oben) erreicht werden.

Beispiele:

`IR 9999,CFS.DAT1`
Es werden maximal 9999 Sätze der Datei CFS.DAT1 zwischen dem ersten und zweiten angezeigten Satz in die Display-Datei eingefügt. Es besteht jedoch die Einschränkung, daß nur soviele Datensätze eingefügt werden, wie freie Schlüssel zwischen dem ersten und zweiten Satz existieren. Gegebenenfalls wird der Einfügevorgang mit DUPKEY-Error abgebrochen.

`IR 20,CFS.DAT2,W`
Angenommen die Referenzdatei CFS.DAT2 enthält nur einen Datensatz, so wird dieser Satz dann 20 mal zwischen dem ersten und zweiten angezeigten Satz der Display-Datei eingefügt.

`IR 999,CFS.DAT3,K`
Angenommen Display-Datei und Referenzdatei CFS.DAT3 sind ISAM-Dateien mit dem gleichen Dateiformat. Es werden soviele Datensätze wie in der Datei CFS. DAT3 vorhanden, maximal jedoch 999 schlüsselgerecht in die Display-Datei übertragen (K-Option). Falls in der Datei bereits Sätze mit entsprechenden Schlüsseln enthalten sind, so werden diese durch die zu übertragenden Sätze überschrieben (nur falls NDUPK-Modus von CFS eingeschaltet ist).

Split (Sätze trennen)

SPLIT [*n*[+|-]], *:col:* [,R]

Beginnend mit dem ersten im Sichtfenster angezeigten Satz werden die nächsten *n* Sätze an der Spalte *:col:* aufgetrennt. Der rechts von *:col:* stehende Teil wird in einen Folgesatz übertragen. Für diesen Satz wird ein neuer Schlüssel generiert.

n Satzanzahl: 1 <= *n* <= 99999.
Das Zeichen '$' anstelle einer Anzahl *n* steht für die maximale Anzahl *n* = 99999.
Standard: *n* = 1. Es wird nur der erste angezeigte Satz getrennt.

n+ Die aufgetrennten Sätze werden mit der kleinstmöglichen Schrittweite beginnend vom unteren Schlüssel an eingefügt.

n- Die aufgetrennten Sätze werden mit der kleinstmöglichen Schrittweite unterhalb des oberen Schlüssels eingefügt.

:col: 1 <= *col* <= 32000
Spalte, nach der der Satz getrennt werden soll. Die Spaltenangabe darf nicht innerhalb des ISAM-Schlüssels liegen.

R Repeat. Das SPLIT-Kommando wird für einen Datensatz solange angewendet, bis alle erzeugten Teilsätze eine Länge kleiner als *col* besitzen.

Beispiele:

```
SPLIT 9999,:88:,R
```
Beginnend mit dem ersten angezeigten Satz an werden die nächsten 9999 Sätze in Teilsätze zu jeweils 88 Bytes (inklusive ISAM-Schlüssel) aufgespalten.

```
SPLIT :258:
```
der erste angezeigte Satz wird bei Spalte 258 in zwei Sätze aufgeteilt.

Join (Sätze zusammenfügen)

JOIN [*n*], {J=*m* | *step*}

n Anzahl der Sätze, auf die das Kommando JOIN anzuwenden ist: 1 <= *n* <= 99999.
Das Zeichen '$' anstelle einer Anzahl *n* steht für die maximale Anzahl *n* = 99999.
Standard: *n* = 1. Es wird nur ein Satz bearbeitet.

Das Join-Kommando ist die Umkehrung des Split-Kommandos. Je nach Angabe des Parameters J=*m* bzw. *step* erfolgt eine verschiedene Verarbeitung.

J=*m* Es werden jeweils *m* aufeinanderfolgende Sätze zu einem einzigen Satz verbunden.

step Beginnend mit dem zweiten im Sichtfenster angezeigten Satz werden die nächsten *n* Sätze in folgender Weise bearbeitet: Liegt der ISAM-Schlüssel des Satzes innerhalb der Step-Size (siehe unten), so wird er an den vorhergehenden Satz angefügt.
step = 0 | 10 | 100 | 1000 | 10000
Als Schrittweite muß der Wert angegeben werden, mit dem die ISAM-Schlüssel zwischen den einzelnen Sätzen ursprünglich erhöht wurden. In diesem Zusammenhang siehe auch Kommando IS auf Seite 293.

Ein Satz wird an den vorhergehenden Satz angefügt, falls sein ISAM-Schlüssel

gleich dem Schlüssel des Vorgängersatzes ist (*step*=0)
in der letzten Stelle einen Wert > 0 und < 10 enthält (*step*=10)
in den letzten 2 Stellen einen Wert > 0 und < 100 enthält (*step*=100)
in den letzten 3 Stellen einen Wert > 0 und < 1000 enthält (*step*=1000)
in den letzten 4 Stellen einen Wert > 0 und < 10000 enthält (*step*=10000).

Liegt ein Satz innerhalb der Step-Size, so ist gewährleistet, daß dies ein durch ein früheres SPLIT-Kommando (siehe oben) erzeugter Folgesatz ist. Die Zuordnung zu dem entsprechenden Hauptsatz ist damit festgelegt.

Hinweis:

Bei der Bearbeitung einer durch das Kommando SPLIT in kürzere Sätze aufgeteilten ISAM-Datei im EDT oder im CFS-Editor ist zu beachten, daß die ISAM-Schlüssel nicht verändert werden dürfen, daß keine neuen Sätze eingefügt und keine vorhandenen Sätze gelöscht werden dürfen. Nur wenn diese Bedingungen erfüllt sind, ist ein korrektes Zusammenfügen der gesplitteten Sätze durch das Kommando JOIN gewährleistet.

Display und Editieren mit SHARUPD=YES

Dateien werden im Display-/Modify-Modus normalerweise mit SHARUPD=NO bearbeitet. Durch das Kommando **SR** (Single Record) wird die Display-Datei mit SHARUPD=YES eröffnet. SR bewirkt außerdem, daß jeweils nur ein Datensatz/Block im Display angezeigt wird. Die Integrität der modifizierten Daten ist damit auch bei konkurrierendem Zugriff durch mehrere Benutzer gewährleistet.

Bei Display-Dateien mit Fcbtype=ISAM/PAM wird der Open mit SHARUPD=YES nur durchgeführt, falls die Datei im normalen Display-Modus (SHARUPD=NO) wegen eines DVS-Fehlers nicht eröffnet werden konnte. Wird die Datei im normalen Display-Modus bereits ohne Fehler angezeigt, so wird der Sharupd-Modus nach Eingabe des Kommandos SR erst aktiv, nachdem der Benutzer mit dem Kommando "/" in den Kommando-Modus des BS2000 gewechselt und mit /R wieder nach CFS zurückgekehrt ist.

Durch Ausschalten des Single Record-Modus mit dem Kommando NSR wird die Display-Datei wieder mit SHARUPD=NO eröffnet.

Falls der Benutzer jedoch die Display-Datei auch mit SHARUPD=YES im ganzseitigen Modus anschauen und ggf. ändern möchte, so ist das Kommando **NSR SHARUPD=YES** einzugeben.

Der SR-Modus gilt nur für die aktuelle Display-Datei und muß bei einer neuen Datei gegebenenfalls wieder gesetzt werden.

Editieren sequentieller Dateien/Bibliothekselemente

Mit dem Modify-Kommando kann jede SAM-Datei/jedes sequentiell organisierte Bibliothekselement 1 zu 1 modifiziert werden. Das Kopieren, Einfügen und Löschen von Sätzen, sowie das Einfügen und Löschen von Spalten ist in diesem Modus bei sequentiellen Daten nicht möglich. Um die erweiterten Editier-Operationen auch auf sequentielle Datenobjekte anwenden zu können, sind die Datenobjekte zuerst in eine ISAM-Hilfsdatei umzusetzen. Die Umsetzung erfolgt mit dem Kommando IS[AM]. Die entsprechende Datei muß jedoch bereits im Display angezeigt werden (Action-Code D). Das Kommando IS bewirkt automatisch die Einschaltung des Modify-Modus für die ISAM-Hilfsdatei.

Die Hilfsdatei kann mit den im letzten Abschnitt beschriebenen ISAM-spezifischen Kommandos bearbeitet werden.

Vor dem Verlassen des CFS-Editors (Kommando LST/Kommando SAM/Drücken der K1-Taste) wird die Hilfsdatei wieder in eine sequentiell organisierte Datei des ursprünglichen Formats zurückkonvertiert, falls der Benutzer auf eine entsprechende Anfrage mit Ja antwortet. Die ISAM-Hilfsdatei wird anschließend gelöscht. Das automatische Löschen der Zwischendatei wird mit dem Kommando SAM,KEEP verhindert. Das genannte Kommando spielt die editierte Zwischendatei ebenfalls in die sequentielle Datei zurück, verhindert aber, daß diese anschließend gelöscht wird.

Display von Banddateien

Im CFS-Display können auch katalogisierte Banddateien mit FCBTYPE=SAM angezeigt werden. Hierzu ist zunächst der Katalogeintrag der gewünschten Banddatei in der CFS-Dateienliste zu selektieren, z.B. über FCBTYPE :TAPE. Bei der katalogisierten Banddatei ist sodann der Action-Code D einzutragen. Das entsprechende Magnetband wird damit an der Konsole zum Montieren angefordert.

Nicht katalogisierte Banddateien können ebenfalls mit dem Action-Code Display angezeigt werden. Es ist dazu folgende Vorgehensweise notwendig:

1) `/FILE datei,LINK=DISP,FCBTYPE=SAM,VOLUME=...,`
   `DEVICE=..,STATE=FOREIGN`

2) Namen der im File-Kommando angegebenen und damit katalogisierten Banddatei in der Dateienliste selektieren. Am einfachsten geschieht dies durch Eintragen des im FILE-Kommando angegebenen Dateinamens im Feld FILENAME-SELECT. Bei der Datei in der Dateienliste Action-Code D eintragen.

Im Normalmodus des Display wird das Band bei jeder Maskenausgabe, bei dem ein bereits gelesener Satzbereich wieder benötigt wird (z.B. Kommandos H, NH, DS, DW) zurückpositioniert und von Anfang an gelesen. Das Zurückpositionieren des Bandes wird nicht durchgeführt, wenn der SR-Modus (Single Record) eingeschaltet ist.

Mit dem Kommando W (Write) bzw. S...=W (Suchen mit Wegschreiben) wird eine Auswahl von Datensätzen der Banddatei auf einfache Weise in eine Plattendatei übertragen.

Editieren von Banddateien

Auch eine Banddatei mit FCBTYPE=SAM (Recform=V/F/U) kann nach Umsetzung in eine ISAM-Hilfsdatei editiert werden.

Vorgehensweise:

1) Falls die Banddatei noch nicht katalogisiert ist, File-Kommando absetzen:
   `/FILE datei,LINK=DISP,FCBTYPE=SAM,VOLUME=...,`
   `DEVICE=..,STATE=FOREIGN`

2) Banddatei über die Selektionsmaske selektieren und in der Dateienliste mit Action-Code D markieren.

3) Banddatei mit Kommando IS[AM] in ISAM-Hilfsdatei umsetzen.

4) Die erzeugte Hilfsdatei wird automatisch im Modify-Modus angezeigt und kann somit mit allen Editieroptionen bearbeitet werden.

5) Nach Abschluß des Editiervorgangs ISAM-Hilfsdatei in Banddatei des ursprünglichen Formats umsetzen: Kommando SAM bzw. K1-Taste.

9. Connections

Connection-Handler (Nebenprozesse unter CFS): Allgemeine Bemerkungen

Der Connection-Handler ist eine Einrichtung, die es ermöglicht, an einem Datensichtgerät mehrere Aufgaben parallel auszuführen, wobei jede Aufgabe für sich einen eigenen Bildschirm erfordern würde.

Connections sind DCAM-Anwendungen in CFS, die z.B. mit $DIALOG oder UTM-Anwendungen kommunizieren.

Für den Benutzer gestaltet sich der Dialog über eine CFS-Connection in der gleichen Weise wie bisher gewohnt. Lediglich das Anmeldeverfahren an eine Connection in CFS unterscheidet sich von der bisherigen Art der Anmeldung. Anstelle des Open-Kommandos im Predialog ist bei CFS im Kommandofeld ein Open-Connection Kommando (OC ...) anzugeben.

Durch den Connection-Handler stehen dem Benutzer folgende zusätzlichen Dienste zur Verfügung:

1) Von einem Datensichtgerät aus können mehrere Dialoge parallel gefahren werden.
2) Aufzeichnen (Logging) und wieder ablaufen lassen (Restore) von Dialogen.
3) Regressionstest (Vergleich des aktuellen Dialogs mit einem aufgezeichneten Dialog).
4) Simulation aufgezeichneter Dialoge.
5) Maskengerechter Hardcopy eines Dialoges (auch mit UTM-Anwendung) in eine druckaufbereitete Datei.
6) Kommandogedächtnis: Aktivieren früherer Benutzereingaben.
7) Parameter-Prompting in Form einer Maske beim Starten von DO-Prozeduren.

Die Administration der verschiedenen Dialoge erfolgt durch die Kommandos OC (Open Connection), CC (Continue in Connection) und DC (Disable Connection).

Pro Benutzer können bis zu 10 Connections eröffnet werden (OC0 bis OC9).

Im folgenden steht *n* für eine Connection-Nummer 0 bis 9.
Statt OC*n*... und CC*n* kann auch die Kurzform *n*... bzw. *n* eingegeben werden. Wenn die Connection noch nicht eröffnet ist, wird das Kommando OC*n* ausgeführt, ansonsten CC*n*.

Nach erfolgreichem **Open Connection** steht der Benutzer in Kommunikation mit der gewünschten Anwendung bzw. es ist ein Dialogprozeß mit einer eigenen TSN unter der angegebenen Benutzerkennung erzeugt worden. Die Kommunikation mit der Connection erfolgt über CFS. Eine Beendigung des Programms CFS hat zur Folge, daß alle Connections verloren gehen.

Soll die Kommunikation mit einem Nebenprozeß zeitweilig unterbrochen werden, z.B. nach dem Start eines Programms oder einer DO-Prozedur, so ist hierzu die **K2-Taste** zu drücken. Der Benutzer kann nun in den Basisprozeß von CFS bzw. in eine andere Connection wechseln. Aus dem CFS-Basisprozeß kann der Benutzer jederzeit den Dialog mit der durch OC*n* eröffneten Anwendung wieder auf-

nehmen durch Eingabe des Kommando CC*n* (**Continue in Connection**). Die in der Zwischenzeit angefallenen Nachrichten werden nach dem CC-Kommando gesammelt an der Datensichtstation ausgegeben.

$CONSOLE: Bei Verbindungen zu $CONSOLE werden alle anfallenden Konsolmeldungen als freilaufende Nachrichten am Bildschirm ausgegeben. Der Benutzer kann Eingaben an die Konsole tätigen, indem er die K2-Taste drückt und auf die danach erscheinende Aufforderung das gewünschte Kommando eingibt. Ein Konsolkommando wie z.B. STA MSG kann sowohl in der Form /STA MSG, als auch in der abgekürzten Form STA MSG eingegeben werden. Zur Änderung der Überlaufsteuerung kann in einer Verbindung zu $CONSOLE auch das Kommando /TCHNG OFLOW= oder /MODIFY-TERMINAL-OPTIONS angegeben werden.

Besonderheiten in CFS-Connections:

Nachrichten, deren Ausgabe durch die Meldung "Please Acknowledge" unterbrochen wird, müssen innerhalb des vom Systemverwalter durch das BS2000-Kommando /BCTIMES LETT=nnn gesetzten Zeitlimits vollständig am Bildschirm ausgegeben werden. Ist dies nicht der Fall, so wird die Verbindung abgebaut (Connection Lost).

Nach dem Verlassen von Programmen, die mit Masken arbeiten (WRTRD MODE=FORM/PHYS) kann es zu Problemen bei der nachfolgenden Eingabe von BS2000-Kommandos kommen. Die Ursache hierfür ist, daß ein Teil der letzten Maske noch am Bildschirm steht. Das Problem kann durch Drücken der Taste Shift Reset Bild (Löschen Bildspeicher) einfach umgangen werden

Die Eingaben LOG / NLOG / RES / HC / NHC / - / -- / ^ werden von CFS als interne Kommandos interpretiert und daher nicht an den Kommunikationspartner übergeben.

Connection eröffnen (Open Connection)

[OC] *n* [*appl*] [, *trm*] [, *host*] [, *opt*] [/*cmd*]
Die Angabe OC ist wahlweise und kann auch weggelassen werden.

appl Name der Anwendung.
z.B. $DIALOG, $CONSOLE oder Name einer UTM-/DCAM-Anwendung.
Standard: $DIALOG (BS2000-Dialogprozeß).

UTM-Anwendung:

In der Generierung der UTM-Anwendung sind für den Anschluß über CFS zusätzliche PTERM- und LTERM-Anweisungen in die KDCDEF aufzunehmen.

Format der PTERM-/LTERM-Anweisungen für UTM:

PTERM *trm*, **PTYPE=***pt*, **PROTOCOL=**N, **PRONAM=***host*, **LTERM=***trm*
LTERM *trm*

trm frei wählbarer virtueller Terminalname, den der Benutzer im OC-Kommando beim Anschluß an die UTM-Anwendung angeben muß. Beispiel: CFS1

host BCAM-Name des Hostrechners, unter dem CFS aufgerufen wurde. Der Hostname wird im Kopf der CFS-Maske in der Spalte HOST = ... angezeigt.
Der Hostname darf nicht mit dem Namen des Vorrechners (siehe STATUS-Kommando) verwechselt werden.

pt Terminaltyp. Es kann hier z.B. T9750 angegeben werden.

Freilaufende Ausgaben von UTM sind mit der Datenübertragungstaste **Shift ENTER** (DUE2) zu quittieren. Diese Taste darf nicht für Verarbeitungsfunktionen der UTM-Anwendung benutzt werden. Falls dies dennoch erforderlich sein sollte, so muß die Datenübertragungstaste für **freilaufende Ausgaben** umdefiniert werden durch das Kommando M L'2307',X'*nn*'. X'*nn*' ist der ASCII-Code für die neu festzulegende Datenübertragungstaste. M L'2307',X'3A' legt z.B. die Shift ENTER-Funktion (DUE2) auf die Taste K14. Näheres zu den Codes für die Datenübertragungstasten finden Sie im Manual "TRANSDATA, Datensichtstation 9750, Schnittstelle für Programmierer".

Falls der Benutzer die Connection zur UTM-Anwendung mit der **Option ES** eröffnen möchte (OC*n appl,trm,*[*host*],ES), d.h. falls die Connection wie ein physikalischer Bildschirm behandelt werden soll, so müssen folgende PTERM-/ LTERM-Anweisungen angegeben werden:

PTERM *trm*,**PTYPE=***pt*,**PROTOCOL=**STATION,**PRONAM=***host*,**LTERM=***trm*
LTERM *trm*

Es ist darauf zu achten, daß die PTYPE-Angabe *pt* übereinstimmt mit dem in der PDN-Generierung festgelegten Terminaltyp des Bildschirms, an dem CFS aufgerufen wurde. Weitere Informationen zur ES-Option finden Sie bei der Beschreibung des Parameters *opt* (Zusatzinformationen für den Aufbau der Verbindung).
Im ES-Modus müssen freilaufende Nachrichten nicht mit Shift ENTER (DUE2) quittiert werden.

Terminal-Pools: UTM bietet die Möglichkeit, mit einer Anweisung TPOOL mehrere PTERMs am gleichen Prozessor zu definieren. Beim Verbindungsaufbau ordnet UTM der aktuellen Connection den nächsten freien PTERM-/LTERM-Namen zu, sofern die Maximalanzahl der im Pool generierten Verbindungen noch nicht erschöpft ist. Im OC-Kommando zur Eröffnung der Connection kann für den Terminalnamen der von CFS

verwendete Standardname benutzt werden (OC*n* *appl*,, [*host*] [,ES]). Beispiel für eine TPOOL-Anweisung:

TPOOL LTERM=*trm***,PROTOCOL=**N|STATION**,PTYPE=***pt***,PRONAM=***host***,
NUMBER=***zahl*

trm Konstantes Prefix für die LTERM-Namen, z.B. LTERM=CFS. Die realen LTERM-Namen werden durch Anhängen einer fortlaufenden Nummer an das Prefix gebildet. Bei der Verwendung der TPOOL-Anweisung sind keine expliziten LTERM-Anweisungen notwendig.

zahl maximale Anzahl der Verbindungen, die von diesem Terminal-Pool bearbeitet werden können.

PROTOCOL=STATION: Das OC-Kommando muß in der Form OC*n* *appl*,,[*host*],**ES** angegeben werden.

PROTOCOL=N: Das OC-Kommando muß in der Form OC*n* *appl*,,[*host*] angegeben werden.

Die Transaktionscodes LOG../NLOG../HC../NHC../RES../O [ON|OFF] werden von CFS intern interpretiert und nicht an die Anwendung gesendet. In diesem Zusammenhang siehe auch Kommando NEC*n* (No Exit for Connection) Seite **376**, sowie CRD (Connection Command Redefine) Seite **375**.

DCAM-Anwendung:

Die Generierung der virtuellen CFS-Terminals ist - falls für die betreffende Anwendung überhaupt erforderlich - nach den internen Konventionen der DCAM-Anwendung vorzunehmen.

Bei der Eröffnung einer Connection zu einer Anwendung, die die CCB-Eigenschaft EDIT=SYSTEM verlangt, muß beim OC-Kommando die Zusatzoption ES angegeben werden.

Bei der Eröffnung einer Connection zu einer Anwendung, die die RPB-Eigenschaft OPTCD=(CS,SPEC) verlangt, muß beim OC-Kommando die Zusatzoption S angegeben werden.

Bei DCAM-Anwendungen, die eine Verbindungsnachricht erwarten, muß beim OC-Kommando die Zusatzoption C angegeben werden.

Eine genaue Beschreibung der Zusatzoptionen des OC-Kommandos folgt weiter unten.

SAP-Anwendung:

Bei SAP-Anwendungen (DCAM) ist die Zusatzoption SAP anzugeben. Der Zusatz SAP darf nicht angegeben werden, falls es sich um eine SAP Anwendung unter UTM handelt. Im Anschluß an den Zusatz SAP kann noch angegeben werden:

TTYPE=... [/NP]

Terminaltyp, den SAP emulieren soll, z.B. SAPTTYPE=T9756. Standardmäßig wird T9750 emuliert. Der wahlweise Zusatz /NP bewirkt, daß von SAP keine P-Tasten oder nur die Tasten P1 bis P4 geladen werden. Bei Angabe von /NP ist zu beachten, daß der Terminaltyp Txxxx acht Byte lang ist, d.h. vor /NP müssen drei Blanks stehen. Beispiel: SAPTTYPE=T9756⊔ ⊔ ⊔/NP.

Anwendung $CONSOLE:

Bei Eröffnung einer Connection zu $CONSOLE wird als Verbindungsnachricht der Name eines berechtigten Benutzerprozeßes (User-Id ohne $ und Punkt), sowie gegebenenfalls ein Passwort (LOGON-Passwort der User-Id) angefordert "Please enter Connection-Message". Die entsprechende Eingabe erfolgt in der Form: *user-id* [,C|X'*passw*']. Zur Definition des berechtigten Benutzerprozeßes sind bei der BS2000-Systemgenerierung UCC-Anweisungen vorzusehen. Für nähere Informationen hierzu siehe Beschreibung "BS2000-Systemgenerierung/Systeminstallation". Ein Beispiel, wie prozedurgesteuert Kommandos an die Konsole abgesetzt werden können, finden Sie auf Seite **510**.

Bis BS2000 V10.0 (siehe vorhergehenden Abschnitt) war es nur möglich, den Zugang zu $CONSOLE über die Definition von berechtigten Benutzerprozessen in der Systemgenerierung (UCC-Anweisungen und starre Verknüpfung zu User-IDs) zu regeln. Die Berechtigungen der einzelnen Konsolen sind somit bereits in der Systemgenerierung festgelegt. Es handelt sich also um "generierte" Berechtigungsnamen.

Ab BS2000/OSD V1 ist es möglich, bei der Anmeldung an $CONSOLE dynamische Berechtigungen zu verwenden. Nach der Anmeldung an die Anwendung $CONSOLE als logische Konsole mit dynamischem Berechtigungsnamen wird mit dem Kommando REQ-OP-ROLE die Berechtigung für eine bestimmte Menge von Kommandos und Routing-Codes angefordert. Die Anmeldung an $CONSOLE erfolgt über ein neues Format der Verbindungsnachricht.

In CFS ist hierzu nach dem Eröffnen der Connection mit OC*n*$CONSOLE bei der Anforderung der Connection-Message folgendes anzugeben:

$CONSOLE | @CONSOLE , *user-id* [, *passwort*] [, *protv*] [, DISCON=YES|NO]

Mit dem Parameter *protv* kann eine Protokollversion für die Konsolapplikation angegeben werden, die den zusätzlichen Austausch von Daten zwischen den Verbindungspartnern festlegt (siehe auch User-Makros NBBME, NBMAP). Die möglichen Werte sind: V00 | V01 | V02 .

Mit dem Parameter DISCON=YES/NO kann das Verhalten von $CONSOLE festgelegt werden, wenn anstehende Nachrichten nicht rechtzeitig von CFS abgeholt werden können.
DISCON=YES: Die $CONSOLE Connection wird beendet.
DISCON=NO: Der Nachrichtenstau wird aufgelöst, indem $CONSOLE die Nachrichten verwirft und stattdessen eine NBR0601-Nachricht ausgibt.

Mit $CONSOLE erfolgt die Anmeldung an das Operating als Terminal, mit @CONSOLE als Programm. Beide Möglichkeiten sind für den Benutzer absolut gleichwertig. Ein Unterschied besteht systemseitig darin, daß mit dem Parameter OPERATOR-ACCESS (siehe weiter unten) für eine User-ID der Konsolzugang lediglich für Terminals ($CONSOLE) oder Programme (@CONSOLE) zugelassen werden kann.

Eine Beschreibung der verschiedenen Möglichkeiten, mit dem Softwareprodukt SECOS die Eigenschaften der Benutzerkennung für den Konsolzugang zu definieren, wird auf das Manual "JES-Benutzerhandbuch", USER OPTION: Zugang zu $CONSOLE verwiesen.

trm Name eines virtuellen Terminals, unter dem die Connection eröffnet wird.
Bei UTM-, sowie bei den meisten DCAM-Anwendungen, muß der virtuelle Terminalname *trm* der Anwendung bekannt sein (Generierung).

Standard: CFS*ntttt* (*n*: Nummer der Connection, *tttt*: TSN des Prozeßes, unter dem CFS geladen wurde).

Der Systemverwalter kann die Art, in der die Standard Terminalnamen von CFS gebildet werden, durch einen alternativen Algorithmus ersetzen. Der Alternativ-Algorithmus bewirkt, daß der Name der physischen Datensichtstation bis auf eine geringfügige Modifikation auch für die Terminalnamen der Connections verwendet wird. Dieses Feature ist über den USERMOD von CFS aktivierbar. Mehr Hinweise hierzu finden Sie im CFS-Handbuch für den Systemverwalter, Abschnitt "Administrationsschnittstellen".

Auf die für **LOGON-Exits** bedeutsame Variante des Status-Kommandos: /STA *tsn*, **TERM=ORIG** sei an dieser Stelle ebenfalls hingewiesen. Der Zusatz TERM=ORIG bewirkt, daß neben dem von CFS oder dem Benutzer vergebenen virtuellen Terminalnamen der Connection auch der Name des Origianalterminals, von dem CFS gestartet wurde, ausgegeben wird (O_STATION, O_PROC).

host Name des Host-Rechners, auf dem die Anwendung läuft.
Bei Angabe von '?' anstelle des Host-Namens wird dieser vom Benutzer in einer separaten Anfrage gepromptet. Anwendungsfall: Menue-System für Connections: Gleichartige Verbindung zu wechselnden Hosts.
Standard: eigener Host.

opt **Zusatzoptionen** für den Aufbau der Verbindung. Es können mehrere Optionen angegeben werden. Die einzelnen Optionen sind durch Kommas zu trennen.
Standard: Es ist keine der beschriebenen Zusatzoptionen aktiviert.

R Rücksprungschalter. Nach erfolgreicher Eröffnung der Connection wird sofort in die Grundebene von CFS verzweigt. Diese Option ist besonders für die prozedurgesteuerte Eröffnung von Connections zu $CONSOLE wichtig.

C Connection-Message. Vor dem YOPNCON wird vom Benutzer eine **Verbindungsnachricht** angefordert. Die dunkelgesteuerte Eingabe kann in der Form *xxx* (formatfreie Eingabe) oder X'...' (Hexadezimal-Format) getätigt werden.

Bei Nichtangabe der C-Option wird von CFS eine **Standard-Verbindungsnachricht** generiert. Falls beim YOPNCON-Aufruf ausdrücklich keine Verbindungsnachricht angegeben werden darf (z.B. DCAM-Anwendung ROUTE), so ist der Parameter NC (siehe unten) zu verwenden.

Bei Eröffnung einer Connection zu $CONSOLE wird als Verbindungsnachricht der Name eines berechtigten Benutzerprozesses (User-Id und LOGON-Passwort) angefordert: "Please enter Connection-Message". Die Eingabe erfolgt in der Form: *xxx*, C'...'. Zur Definition eines berechtigten Benutzerprozeßes sind bei der Systemgenerierung UCC-Anweisungen vorzusehen. Für nähere Informationen hierzu siehe Beschreibung "BS2000-Systeminstallation".

NC Beim Verbindungsaufbau mit YOPNCON wird keine Verbindungsnachricht generiert.

UC Konsolartige Verbindung. Freilaufende Ausgaben im Line-Modus, Eingabemöglichkeit nur nach Betätigen der K2-Taste.

UCH wie UC, jedoch werden den Eingaben Subsystem-Header vorangestellt.

HC Nur bei konsolartiger Verbindung: Freilaufende Ausgaben im Line-Modus werden an einem lokal angeschlossenen Hardcopy-Drucker protokolliert.

U UTM-artige Verbindung (strenger Dialog im Format-/Physical-Modus).

U1 UTM-artige Verbindung. Diese Kompatibilitätsoption ist anzugeben, um folgende Situation zu vermeiden: UTM-Masken werden nicht vollständig angezeigt. Es wird an einem gelöschten Bildschirm nur die letzte Teilmaske ausgegeben. CFS löscht den Bildschirm bei einer vorausgehenden Nachricht, die eine Auto-DUE Befehlsfolge (PAR00D-Steuerblock) enthält.

S CFS schließt sich an die Anwendung mit den Attributen OPTCD=(CS,SPEC) an. Manche DCAM-Anwendungen, z.B. SAP akzeptieren nur Partner mit OPTCD=(CS, SPEC). Standard: OPTCD=(CA,ANY).

A3 A3NOP kompatibler Modus. In der als erstes ausgegebenen Bildschirmmaske kann der Benutzer Eingaben tätigen (WRTRD-Aufruf). Alle folgenden Nachrichten werden vom Connection-Handler als freilaufende Ausgaben behandelt (WROUT-Aufruf). Eine mit der A3-Option eröffnete Connection kann nur in der Grundebene von CFS mit dem DC-Kommando geschlossen werden. Der A3-Modus ist geeignet für Connections zu Anwendungen wie A3NOP, die freilaufende Ausgaben erzeugen.

ES | ES1 | ES2 | ESO
EDIT=DSSIM: Mit dieser Option wird erreicht, daß sich die Connection gegenüber dem Betriebssystem wie ein realer physikalischer Bildschirm verhält. Siehe hierzu auch Seite 345 "Maßnahmen bei nicht erfolgreichem Connection-Aufbau". Im ES-Modus müssen freilaufende Nachrichten nicht mit Shift ENTER (DUE2) quittiert werden.

ES1 | ES2 ES1 und ES2 sind besondere Varianten des ES-Modus für DCAM-Anwendungen wie OMNIS, SILINE, RDDIA, FTDIA. ES1 und ES2 unterscheiden sich nur geringfügig in der Behandlung der gesendeten K2-Nachrichten.

ESO Edit System with Original Terminal-name: Diese Option kann nur für eine Connection angegeben werden. Sie bewirkt, daß als virtueller Terminalname der Connection der Name des physischen Bildschirms, an dem CFS gestartet wurde, verwendet wird.

NES Über den Konfigurationsmodul von CFS kann eingestellt werden, daß alle Connections zu $DIALOG automatisch im ES-Modus eröffnet werden.
Mit der Option NES kann in diesem Fall eine Nicht-ES Connection eröffnet werden.

LC Lower Case. Nur von Bedeutung bei Connections zu $VMCONS: Es werden Kleinbuchstaben an die Anwendung übergeben. Normalerweise werden bei konsolartigen Verbindungen eingegebene Kleinbuchstaben in Großbuchstaben umgewandelt.
Achtung: Diese Option darf auch bei VMCONS-Connections nur für die Administration einer Linux-VM angegeben werden.

DCAM8 Die Verbindung wird im ACB (DCAM Application Control Block) mit DCAMVER=8.0 eröffnet. Standard: DCAM 7.0. Dieser Parameter ist notwendig, falls der Primärprozess ebenfalls die Angabe DCAMVER=8.0 verwendet.

T9763 | T9755 | T9750
Es wird eine Dialog Connection mit dem angegebenen Terminaltyp eröffnet. Der Zusatz kann nur in einem Batch-Lauf von CFS angegeben werden.
Beispiel: OC1,,,T9763,ES/LOGON .....
Damit kann in einem Enter-Prozeß eine 8-bit Terminalconnection eröffnet werden. In der Connection kann mit /MOD-TERM-OPTION CODED-CHAR=EDF041 ein 8-bit Zeichensatz eingestellt werden.

SAP [[TTYPE=...] [/NP]]
Die angegebene Anwendung ist eine SAP DCAM-Anwendung. Es wird für SAP die Standard-Verbindungsnachricht 'TTYPE=T9750␣␣␣/NC' bereitgestellt. Der Systemverwalter kann in CFSMAIN einen vom Standard-Terminaltyp T9750 abweichenden Terminaltyp angeben. SAP entspricht der Option U,S, wobei zusätzlich eine Standard-Verbindungsnachricht generiert wird. Die Option SAP darf nicht angegeben werden, falls es sich um eine SAP-Anwendung unter UTM handelt. Im Anschluß an den Zusatz SAP kann noch angegeben werden:

TTYPE= Terminaltyp, den SAP emulieren soll, z.B. SAPTTYPE=T9756. Standardmäßig wird T9750 emuliert.

/NP Der Zusatz /NP bewirkt, daß von SAP keine P-Tasten oder nur die Tasten P1 bis P4 geladen werden. Bei Angabe von /NP ist zu beachten, daß der Terminaltyp Txxxx acht Byte lang ist, d.h. vor /NP müssen drei Blanks stehen.
Beispiel: SAPTTYPE=T9756⊔ ⊔ ⊔/NP.

/cmd Es wird ein BS2000-Prozeß zu $DIALOG eröffnet und danach das angegebene BS2000-Kommando ausgeführt. Es sind zwei Fälle zu unterscheiden:

a) */cmd* = /LOGON *user-id,acct*,C'...'
Der Dialogprozeß wird unter der angegebenen Benutzerkennung eröffnet.

b) */cmd* = beliebiges BS2000-Kommando außer LOGON.
Es wird ein Dialogprozeß unter der eigenen Benutzerkennung (d.h. unter der Benutzerkennung, unter der CFS aufgerufen wurde) eröffnet. Als erstes wird das angegebene BS2000-Kommando in dem so eröffneten Dialogprozeß ausgeführt. Falls die eigene Benutzerkennung ein LOGON-Passwort besitzt, so wird dieses während des Aufbaus der Verbindung vom BS2000 nachgefordert.

Beispiele:

`OC1`
Open Connection 1. BS2000-Dialogprozeß am eigenen Host-Rechner eröffnen. Im Anschluß an das Kommando OC1 erscheint die normale BS2000-Logon Aufforderung ".... PLEASE LOGON ...". Der Benutzer kann daraufhin ein LOGON-Kommando für eine beliebige Kennung eingeben.

`OC1,,host/LOGON user-id,acct,C'passw'`
Dialogprozeß am angegebenen Host-Rechner als Connection 1 eröffnen. Das LOGON-Kommando wird in diesem Beispiel bereits im OC-Kommando angegeben.

`OC9/EXEC $EDT`
Dialogprozeß (Connection 9) am eigenen Host-Rechner und unter der eigenen Benutzerkennung eröffnen. Nach erfolgreichem LOGON und nach Beantwortung des Bulletins wird automatisch das BS2000-Kommando /EXEC $EDT ausgeführt.

`OC2 TESTUTM,CFS1`
Connection 2 zu der Anwendung TESTUTM eröffnen. Der virtuelle Terminalname für diese Connection lautet CFS1. In der KDCDEF für die Anwendung TESTUTM muß eine PTERM- und LTERM-Anweisung für den Namen CFS1 vorhanden sein. Im Anschluß an die Eröffnung dieser Connection erscheint die UTM-Aufforderung zur Eingabe eines KDCSIGN-Kommandos.

`OC3 SAP,,VAR3,SAP`
Unter der Nummer 3 wird eine Verbindung zu der DCAM SAP-Anwendung SAP eröffnet. Die Anwendung befindet sich auf dem Host VAR3 und läuft nicht unter der Steuerung von UTM. Die Zusatzoption SAP teilt CFS mit, daß es sich hier um eine Verbindung zu einer SAP-Anwendung handelt.

`OC4 UDSDB,,,UC`
Unter Nummer 4 wird eine Verbindung zu der DCAM-Anwendung für die UDS-Administration eröffnet. Bei der von CFS angeforderten Connection-Message ist das Administrationspaßwort, soweit notwendig, im Klartext anzugeben. Z.B. ist bei einem Passwort der Form C'ABC' lediglich ABC einzugeben. Nach erfolgtem Verbindungsaufbau kann durch Betätigen der K2-Taste ein Kommando an das Administrationsprogramm übergeben werden.

Maßnahmen bei nicht erfolgreichem Connection-Aufbau

Als erstes ist der **DCAM-Returncode** (angezeigt in der Hinweiszeile der CFS-Maske: R15=X'.....' / R6=X'.....') auszuwerten. Eine Auflistung aller Returncodes finden Sie in den Anhängen A-2/A-3 des Manuals "DCAM-Makroaufrufe". Falls der DCAM-Returncode keinen Hinweis auf die Fehlerursache liefert, so sind die in der folgenden Checkliste aufgeführten Punkte zu prüfen.

UTM-Anwendung:

Virtueller Terminalname im OC-Kommando angegeben ?

Verbindungsoption ES versuchen: OC*n appl* [, *term*] [, *host*],**ES**

KDCDEF-Generierung durch neue CFS-Terminals ergänzt (PTERM, LTERM) ? (näheres siehe Seite 339).

Falls UTM-Anwendung an einem anderen Host-Rechner: Korrekter Name des Host-Rechners im OC-Kommando angegeben ? Name des Host-Rechners, auf dem der CFS-Basisprozeß eröffnet ist, in der PTERM-Anweisung der KDCDEF-Datei der UTM-Anwendung korrekt angegeben ?

UTM-Anwendung nicht hochgefahren ? (DCAM-Returncode X'0C4C0000')

DCAM-Anwendung:

OC-Kommando mit einer der Zusatzoptionen ES bzw. ES1/ES2 versuchen (EDIT= DSSIM, Simulation eines realen Bildschirms). Diese Optionen sind z.B. beim Anschluß an die Anwendungen **SILINE, TDSPOOL, RDDIA** notwendig.

OC-Kommando mit Zusatzoption S versuchen (--> OPTCD=(CS,SPEC)).

OC-Kommando mit Zusatzoption DCAM8 versuchen (--> DCAMVER=8.0 im ACB).

Verlangt Anwendung die Angabe einer Verbindungsnachricht ? Eine Verbindungsnachricht wird beim Anschluß von einem physikalischen Terminal in der Form O..., MSG='...' angegeben. Falls Verbindungsnachricht erforderlich:

OC-Kommando mit Zusatzoption C versuchen. Danach Eingabe der Verbindungsnachricht. Die Verbindungsnachricht darf im Gegensatz zu MSG='...' nicht in Hochkommas eingeschlossen werden. Falls keine Verbindungsnachricht verwendet werden darf (z.B. bei Anwendung **ROUTE**), ist die Zusatzoption NC (No Connection Message) zu anzugeben.

Falls alles bisherige nicht zutrifft:

OC-Kommando mit der Zusatzoption A3 (A3NOP-Modus) versuchen.
Der A3-Modus sollte bei Verbindungen zu Anwendungen wie A3NOP, die freilaufende Ausgaben erzeugen, angegeben werden.

OC-Kommando mit der Zusatzoption UC (konsolartige Verbindung) versuchen.
Dies ist z.B. bei **UDS Database-Handlern**, sowie bei der **VM2000-Console** (Anwendung $VMCONS) notwendig.

DCAM-Anwendung nicht hochgefahren ? (DCAM-Returncode X'0C4C0000')

OC-Kommando wurde nicht zurückgewiesen, es kommt jedoch keine Antwort von der Anwendung: Betätigung der K2-Taste + Eingabe B (Break) versuchen.

Kommunikation mit unterbrochenem Dialog aufnehmen (Continue in Connection)

[CC] *n*[/*cmd*]

Continue in Connection *n*

Die Angabe CC ist wahlweise und kann auch weggelassen werden.

Unterbrochenen Dialog mit Verbindung *n* wieder aufnehmen. Es wird aus der Grundebene von CFS in den angegebenen Nebenprozeß verzweigt.

cmd

Falls es sich bei der Connection um einen BS2000 Dialogprozeß handelt, so kann im CC-Kommando eine Eingabe für den Nebenprozeß angegeben werden.
Die Eingabe wird sofort nach Verzweigen in den Nebenprozeß ausgeführt. Falls der Benutzer in der Connection in einer Maske oder Physical-Mode-Eingabe steht, so wird *cmd* solange zurückgehalten, bis die erste Line-Mode-Eingabe vom Partner angefordert wird.

Von CFS zu interpretierende Transaktionscodes wie LOG/NLOG/HC usw. werden hier von CFS nicht erkannt und demnach auch nicht ausgewertet. Bei Eingabe des BS2000-Kommandos DO *proc* kann das Job-Report System von CFS nicht aktiviert werden.

Um die genannten Funktionen nutzen zu können, ist die Eingabe in zwei Schritten erforderlich: 1) CC*n* und 2) *cmd*

Bei Verbindungen zu UTM-Anwendungen können dem CC-Kommando generell keine Eingaben mitgegeben werden.

[CC] *n*B

Continue in Connection *n*. Durch 'B' wird ein Break für diese Connection ausgelöst. Gleiche Wirkung wie Eingabe 'B' in der K2-Anfrage (siehe unten).

[CC]*n*,R*tt*

Bei Connections zu $CONSOLE: Es wird in die Connection *n* verzweigt.
Nach *tt* Sekunden erfolgt automatisch ein Return in die Grundebene von CFS (Kommandofeld). Diese Variante des CC-Kommandos wurde speziell zur Anwendung in CFS-Prozeduren geschaffen.

[CC] *n*$RS

Fehlerbehebung: In seltenen Fällen kann die Fehlersituation eintreten, daß Eingaben doppelt eingegeben werden müssen bzw. mit einer Eingabe Verspätung ausgeführt werden. In dieser Situation können Sie einen Rettungsversuch unternehmen, indem Sie mit K2 und R die Connection verlassen und anschließend mit CC*n*$RS die internen Indikatoren in der Connection zurücksetzen.

Hinweis:

Die Kommandos OC*n*/CC*n* können auch verkürzt eingegeben werden, indem nur die Nummer der gewünschten Connection angegeben wird. CFS ergänzt in diesem Fall CC bei bereits eröffneter und OC bei noch nicht eröffneter Connection.

Dialog mit Connection administrativ schließen (Disable Connection)

DC*n*

Disconnect Connection *n*.

DC wird in der Grundebene von CFS im Kommandofeld eingegeben und bewirkt das Schließen der angegebenen Verbindung. Falls unter der Connection noch ein

BS2000-Dialogtask aktiv war, so hat DC*n* eine Konsolmeldung "ABNORMAL TASK TERMINATION 22" (Leitungsverlust/Hardware-Fehler) zur Folge.

Ein Nebenprozeß wird auch durch das BS2000-Kommando /LOGOFF beendet.
Manche Connections, wie z.B. Consolverbindungen, können jedoch nur mit dem DC-Kommando beendet werden.

K2-Taste in Connections

Die K2-Taste, die normalerweise den unmittelbaren Übergang in die Kommando-Ebene des BS2000 bewirkt, hat in Connections die Funktion, den Übergang zwischen den Connections bzw. in die Grundebene von CFS zu steuern. Nach Drücken der K2-Taste werden folgende Aktionen ausgeführt:

1) Der aktuelle Bildschirminhalt wird sichergestellt.

2) Bei eingeschaltetem ACD-Modus (Automatic Connection Display) werden alle im Moment eröffneten Connections angezeigt (Standard).

3) Es wird eine Zeile ausgegeben, die anzeigt, in welcher Connection sich der Benutzer gerade befindet.

4) Es wird folgende Meldung ausgegeben:
   `Return to CFS/Continue in Conn. n/Break ?  (R/n/n cmd/B/BB/^cmd/?)`

   Bei Connections zu $CONSOLE lautet die Meldung:
   `Console-`
   `Input/Return to CFS/continue in Conn. n ? (input/R/n/^cmd/?)`

   Bei Connections zu $CONSOLE im Asynchron-Modus (AC*n*) lautet die Meldung:
   `Input for Console n/Return to CFS ?  (n input/R/^cmd/?)`

   Bei Connections zu $CONSOLE und $VMCONS hat der Benutzer zwei Minuten (120 Sekunden) Zeit, um obige Anfrage zu beantworten. Ist in dieser Zeit keine Eingabe erfolgt, so wird wieder zur Ausgabe von Konsolmeldungen zurück verzweigt. Im Modul CFSMAIN kann der Systemverwalter die Wartezeit auf einen Wert zwischen 10 und 3600 Sekunden festlegen. Der Eintrag von 0 Sekunden bewirkt, daß wie in früheren Versionen von CFS, unbegrenzt lange auf eine Benutzereingabe gewartet wird.

Folgende Eingabemöglichkeiten stehen zur Wahl:

R
: Rückkehr in den CFS-Grundprozeß. Mit dem Kommando CC*n* kann der Bildschirm wieder auf die Connection mit der Nummer *n* geschaltet werden.

input
: nur bei einer Verbindung zu $CONSOLE: gewünschte Eingabe an die Konsole.
z.B.: *tsn*.INTR, STA MSG,ALL

n input
: nur möglich, falls mehrere $CONSOLE-Verbindungen in den Asynchron-Modus versetzt wurden (Kommando AC*n*): gewünschte Eingabe an die in der Connection *n* eröffnete Konsole. z.B.: 1 *tsn*.INTR, 9 STA MSG,ALL

n
: Wechsel in die angegebene Connection.
n ist die Nummer einer bereits eröffneten Connection. Für *n* kann auch wieder die Nummer der aktuellen Connection angegeben werden.

n cmd
: wie oben, jedoch mit zusätzlicher Eingabe eines Kommandos in der neuen Connection (z.B. 2/STA L). Für *n* kann auch wieder die Nummer der aktuellen Connection angegeben werden.

	cmd wird sofort nach Verzweigen in die Connection *n* ausgeführt. Falls in der Connection eine Format- oder Physical-Mode Eingabe angefordert wird, so wird *cmd* bis zur nächsten Line-Mode Eingabe zurückgehalten.
B	Die K2-Taste wird an die Connection weitergegeben. Bei umfangreichen Ausgaben, wie z.B. Dateiübersichten aufgrund eines FSTAT-Kommandos, kann der Fall eintreten, daß die K2-Taste erst nach mehreren Bildschirmseiten zum BS2000-Kommandoprompt führt.
BB	Die K2-Taste wird an die Connection weitergegeben. Im Unterschied zu B (siehe oben) werden bei umfangreichen Ausgaben, z.B. FSTAT-Kommando alle Nachrichten bis zum nächsten Eingabeprompt verworfen.
^*cmd*	CFS verzweigt in den Grundprozeß, führt das angegebene Kommando (die angegebene Kette von Kommandos) aus und kehrt, soweit möglich, wieder in die aktuelle Connection zurück. Die Wirkung von ^*cmd* ist im nächsten Abschnitt "Fluchtsymbol in Connections" ausführlich beschrieben. Die Eingabe ^2/ bewirkt z.B. das Eröffnen der Connection 2 unter der eigenen Benutzerkennung, bzw. das Verzweigen in diese Connection, falls Connection 2 bereits eröffnet ist.
?	Aktivierung des Help-Systems. Es werden die Eingabemöglichkeiten beschrieben.

Hinweise:

Die bloße Betätigung der ENTER-Taste ohne Eingabe einer Option wirkt wie R (Return to CFS).

Die oben beschriebene Sonderfunktion der K2-Taste in CFS-Connections kann für jede Connection durch das Kommando DKC*n* (Direct K2-mode for Connection *n*) ausgeschaltet werden. Für weitere Informationen siehe Seite **376**.

In Connections zu $DIALOG sind auch Administrationsprogramme ablauffähig, für die normalerweise die K2-Taste betätigt werden muß, um eine Eingabeaufforderung zu erhalten. Um in einer CFS-Connection die Eingabeaufforderung des Programms zu erhalten, ist die K2-Taste zu drücken und bei der danach erscheinenden Frage die Antwort B (Break) einzugeben.

Bei UTM-/DCAM-Anwendungen kann der Fall eintreten, daß die Eingabe **K2** + *n* nicht in die gewünschte Connection mit der Nummer *n*, sondern in die Grundebene von CFS führt. In diesem Fall sollte der Bildschirm nach dem Drücken der K2-Taste mit Shift Reset Bild gelöscht werden, bevor die Connection-Nummer *n* eingegeben wird.

Bei UTM-/DCAM-Anwendungen (z.B. OASE, ADABAS) kann der Fall auftreten, daß nach Drücken der K2-Taste nicht die gewohnte CFS Break-Anfrage erscheint. Es gibt zwei Möglichkeiten, dieses Problem zu umgehen:

- Mit Shift Reset Bild den Bildschirm löschen und danach K2-Taste betätigen
- Verwendung des Fluchtsymbols ^ / ^n, um in den Grundprozeß von CFS oder in eine andere Connection zu wechseln.

Falls in einer Connection zu $DIALOG die Fehlermeldung **"Message cannot be displayed ..."** ausgegeben wird und die Connection danach nicht mehr weiter läuft, so kann zur Wiederherstellung der Eingabeaufforderung die K2-Taste gedrückt werden und die folgende Conn.-Handler Anfrage mit B (Break) beantwortet werden. Dies ist unter Umständen mehrmals zu wiederholen, bis der Schrägstrich ("/") bzw. der SDF-Prompt ("KDO:" / "CMD:") ausgegeben wird.

Die K2-Taste kann jederzeit betätigt werden, z.B. auch während eine DO-Prozedur oder ein Programm in der Connection läuft. Im Gegensatz dazu kann das Fluchtsymbol (siehe folgender Abschnitt) nur bei einer Eingabeaufforderung in der Connection angegeben werden.

Fluchtsymbol in Connections

Bei Betätigung der K2-Taste in Connections wird der Connection-Display sowie eine Zeile mit Angaben über die aktive Connection ausgegeben. Danach kann der Benutzer die für ihn relevante Eingabe tätigen: R (Rückkehr nach CFS)/ *n* (Continue in Conn. *n*) / B (Break).

Der geübte Benutzer kann dieses Verfahren abkürzen durch Eingabe des Fluchtsymbols. Als Fluchtsymbol gilt standardmäßig das Zeichen ^. Bei Connections der Ebene 2 (siehe unten "geschachtelte Connections") wird das Fluchtsymbol auf das alternative Zeichen ` gesetzt.

Das Fluchtsymbol kann sowohl im Line-Modus als erstes Zeichen der Eingabe, als auch im Format-Modus in der ersten beschreibbaren Stelle der Bildschirmmaske (Home-Position) angegeben werden.

Das Fluchtsymbol kann nur auf eine Eingabeaufforderung in der Connection hin angegeben werden. Im Unterschied zum Fluchtsymbol ist die Betätigung der K2-Taste unabhängig von der Eingabeaufforderung in der Connection möglich, z.B. auch während eine DO-Prozedur oder ein Programm läuft.

Zusammen mit dem Fluchtsymbol können folgende Optionen eingegeben werden:

^ — Verzweigen in den Grundprozeß von CFS.

^B — Break für aktuelle Connection. Entspricht der Tastenfolge **K2** + B.

^*n* — Aus der aktuellen Connection in Connection *n* verzweigen.

^*n cmd* — Aus der aktuellen Connection nach Connection *n* wechseln. *n* kann auch die Nummer der gerade aktuellen Connection sein. In der neuen Connection wird als erstes das angegebene Kommando abgesetzt. Entspricht der Tastenfolge **K2** + '*n cmd*'. *cmd* wird sofort nach Verzweigen in den mit *n* spezifizierten Nebenprozeß ausgeführt. Falls der Benutzer in der Connection in einer Maske oder Physical-Mode Eingabe steht, so wird *cmd* solange zurückgehalten, bis die erste Line-Mode Eingabe angefordert wird.

^*cmd* — Es wird in den CFS-Grundprozeß verzweigt. Dort wird das angegebene Kommando /die angegebene Kette von Kommandos (Separatorzeichen = ';') ausgeführt. Nach ordnungsgemäßer Ausführung aller Kommandos wird in der Regel wieder in die aktuelle Connection zurückverzweigt. Bei einigen Kommandos ist diese Regelung nicht sinnvoll. CFS kehrt dann nicht in die Connection zurück (z.B. neue Dateiselektion mit ^NP / Verzweigen in eine andere als die aktuelle Connection mit ^*n* [/*cmd*]).

Bei nicht ordnungsgemäßer Ausführung des Kommandos bleibt CFS im Grundprozeß und kehrt nicht in die Connection zurück.

^ : *x* Umdefinieren des Fluchtsymbols in ein anderes Zeichen.

Hinweise:

Das Fluchtsymbol wird wie eine Benutzereingabe interpretiert und entsprechend ausgeführt, wenn es in einer Bildschirmausgabe im Line-Mode als erstes Zeichen erkannt wird.
Beispiel: /WRITE-TEXT '^4*xxxx*'
Damit ist es z.B. möglich, aus einem Programm oder einer DO-Prozedur, die in einer Connection abläuft, quasi programmgesteuert in eine andere Connection (z.B. Conn. 4) zu wechseln und dort eine Eingabeaufforderung im Line-Modus z.B. mit *xxxx* zu beantworten.

Falls in einer Connection noch einmal CFS geladen wurde und auch dort CFS Connections eröffnet werden (Connections der Ebene 2), so ergeben sich Überschneidungen bei der Verwendung des gleichen Fluchtsymbols für die Connections der Ebenen 1 und 2. In dem in einer Connection aufgerufenen CFS wird daher automatisch das **alternative** Zeichen ` (X'4A') als **Fluchtsymbol** verwendet. Befindet sich der Benutzer in einer Connection der Ebene 2, so gelangt er durch Eingabe von ^ in das als erstes geladene CFS (Ebene 0). Durch Eingabe des alternativen Fluchtsymbols ` gelangt der Benutzer in das in der Connection geladene CFS (Ebene 1). Es ist daher sinnvoll, geschachtelte Connections mit dem alternativen Fluchtsymbol ` zu verlassen.

Das Fluchtsymbol gilt einheitlich für alle Connections einer Schachtelungsstufe.

Der Systemverwalter kann über eine wahlweise Option CFS so generieren (siehe CFSMAIN), daß das Fluchtsymbol vor allen, für den Connection-Handler von CFS bestimmten Kommandos angegeben werden muß. Das Kommandogedächtnis wird dann nicht durch "-", sondern durch "^-" aktiviert. Die LOG-, bzw. RES-Funktion wird in diesem Fall durch ^LOG, bzw. ^RES eingeschaltet. Der Hardcopy-Modus wird durch ^HC aktiviert und durch ^NHC beendet. Ob die ^-Option in CFS aktiviert wurde, kann durch Aufrufen der CRD-Maske (Kommando CRD) erkannt werden.

Geschachtelte Connections

Falls in einer Connection noch einmal CFS geladen wurde und auch dort Connections eröffnet werden (Connections der Ebene 2), so ergeben sich Überschneidungen bei der Verwendung des gleichen Fluchtsymbols für die Connections der Ebenen 1 und 2. In dem CFS, das in einer Connection geladen wurde, wird daher automatisch das alternative Zeichen ` (X'4A') als Fluchtsymbol verwendet. Befindet sich der Benutzer in einer Connection der Ebene 2, so gelangt er durch Eingabe von ^ in das als erstes geladene CFS (Ebene 0). Durch Eingabe des alternativen Fluchtsymbols ` gelangt der Benutzer in das in der Connection geladene CFS (Ebene 1). Es ist daher sinnvoll, geschachtelte Connections mit dem alternativen Fluchtsymbol ` zu verlassen.

Im Zusammenhang mit geschachtelten Connections ist außerdem zu beachten, daß bei Betätigung der K2-Taste in Ebene 2 und anschließender Eingabe R (Return) in das als erstes geladene CFS (Ebene 0) und nicht zur direkt untergeordneten CFS-Ebene 1 verzweigt wird. Ein später folgendes Kommando CC*n* führt nicht in das CFS der Ebene 1, sondern in die zuvor mit **K2** verlassene Connection der Ebene 2. Bei Betätigung der K2-Taste in einer Connection der Ebene 2 und anschließender Eingabe B (Break) meldet sich darauf das CFS in der Connection-Ebene 1 ebenfalls mit seiner Connection-Anfrage "Return / Continue in Conn. n / Break ...".

Bei Absetzen eines HC-, LOG-, RES-Kommandos in einer geschachtelten Connection wird das Kommando stets vom Connection-Handler des CFS bearbeitet, das als erstes geladen wurde (CFS der Ebene 0, Basis-CFS). LOG- und Hardcopy-Dateien werden daher stets in der Benutzerkennung angelegt, in der CFS zum ersten Mal aufgerufen wurde.

In den verschiedenen Connection-Ebenen kann das Symbol zum Aufruf des Eingabegedächtnisses (siehe unten Abschnitt) durch das Kommando CRD (Connection Command Redefine) für jede Ebene unterschiedlich definiert werden. Das CRD-Kommando zum Verändern des Symbols für das Eingabegedächtnis für alle Connections einer Ebene ist in der Kommandozeile des CFS anzugeben, das diese Connections direkt verwaltet, also in der direkt darunter liegenden Ebene.

Die Bearbeitung aller CFS-spezifischen Connection-Kommandos wie HC, LOG, RES, - kann auch für eine bestimmte Connection x der Ebene n unterbunden werden. Hierzu ist in dem CFS der Ebene n-1 das Kommando NEC x (No Exit for Connection x) einzugeben.

Kommandogedächtnis

Der Connection-Handler von CFS besitzt ein Kommandogedächtnis. Alle im Line-Mode und bestimmte im Format-/Physical-Mode getätigten Eingaben für die Connection werden in einem internen Speicher abgelegt. Es werden alle auf der BS2000-Ebene und im EDT eingegebenen Kommandos für eine spätere Wiederverwendung gesichert. Im Kommandogedächtnis gespeicherte Einträge können jederzeit wieder auf den Bildschirm geholt und unverändert oder verändert abgesendet werden. Der Zugriff auf die im Kommandogedächtnis gespeicherten Eingaben ist auf zwei verschiedene Arten möglich:

a) sequentiell:

-◄ ENTER. Im Line-Modus wird das letzte in dieser Connection eingegebene Kommando in der 24-ten Bildschirmzeile angezeigt. Im Anschluß an die Eingaben aus der aktuellen Connection werden auch die Eingaben für alle anderen eröffneten Connections angezeigt.

b) assoziativ:

-*string*◄ ENTER. Es wird das letzte Kommando angezeigt, welches mit der Zeichenfolge *string* beginnt.

-**string*◄ ENTER. Es wird das letzte Kommando angezeigt, welches die Zeichenfolge an beliebiger Stelle enthält.

Das Zeichen ◄ steht hier für die Endemarke (EM).

Durch Betätigen der Taste Shift ENTER (DUE2) wird jeweils das vorletzte, vorvorletzte Kommando usw. angezeigt.

Erscheint die Eingabe, die der Benutzer aktivieren möchte, so ist diese mit der ENTER-Taste abzusenden. Vor dem Absenden kann die am Bildschirm stehende frühere Eingabe auch noch verändert werden. Die Ausgabe des Kommandogedächtnisses wird mit Betätigung der ENTER-Taste beendet. Das Beenden des Kommandogedächtnisses ohne Aktivierung einer Eingabe erfolgt durch Betätigen der K1-Taste.

Die im Kommandogedächtnis abgelegten Eingaben werden wie folgt aktiviert:
Zuerst werden die in der aktuellen Connection eingegebenen Kommandos angezeigt. Bei weiterem Zurückgehen im Kommandogedächtnis werden die Kommandos aus den Connections 0 bis 9 mit Ausnahme der aktuellen Connection angezeigt. Die Kommandogedächtnisse aller Connections sind miteinander verkettet und werden als Einheit betrachtet.

Das Symbol für das Kommandogedächtnis '-' kann vom Benutzer auch umdefiniert werden: *sym-alt* : *sym-neu*. Beispiel: -:]◄ ENTER

Das Umdefinieren des Kommandogedächtnis-Symbols ist z.B. notwendig, falls der Benutzer im Line-Modus des EDT durch Eingabe von '-' zurückblättern möchte. Die Änderung des Symbols für das Kommandogedächtnis gilt global für alle Connections.

Hinweise:

Das nachträgliche Entfernen bestimmter Eingaben aus dem Kommandogedächtnis ist wie folgt möglich: Das Kommandogedächtnis wird aktiviert durch Eingabe von -◄ ENTER. Die zu entfernende Eingabe wird durch sequentielles Zurückgehen mit Shift ENTER (DUE2) aufgesucht und durch Betätigen der K3-Taste aus dem Gedächtnis gelöscht.

Eingaben, die nur aus einem oder zwei Zeichen bestehen, werden nicht im Kommandogedächtnis abgelegt.

Maskeneingaben (Mode=Format/Physical) werden nicht in das Kommandogedächtnis eingetragen. EDT- und KDC-Kommandos werden im Kommandogedächtnis abgelegt.

EDT-Eingaben können über das Kommandogedächtnis aktiviert werden durch Voranstellen eines Leerzeichens:
␣-◄ ENTER | ␣-*string*◄ ENTER | ␣-**string*◄ ENTER.

Das Kommandogedächtnis kann auch bei Verbindungen zu $CONSOLE genutzt werden. Zur Anzeige früherer Konsoleingaben ist die K2-Taste zu drücken und anschließend -◄ ENTER einzugeben.

Im Administrationsdialog von UTM-Anwendungen kann das Kommandogedächtnis ebenfalls genutzt werden. Die Anzeige früherer Eingaben erfolgt durch Eingabe von -◄ ENTER.

Der Systemverwalter kann über eine wahlweise Option CFS so generieren, daß das Fluchtsymbol vor für den Connection-Handler von CFS bestimmten Kommandos angegeben werden muß. Das Kommandogedächtnis wird dann durch "^-" aktiviert. Ob die ^-Option in CFS aktiviert wurde, kann durch Aufrufen der CRD-Maske (Kommando CRD) erkannt werden.

Inhalte aus Kommandogedächtnis in P-Tasten laden

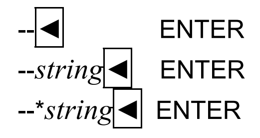
Ein am Bildschirm angezeigter Eintrag aus dem Kommandogedächtnis kann in eine frei bestimmbare P-Taste geladen werden. Hierzu ist die Bezeichnung der P-Taste der Eingabe voranzustellen: 'P*x*#' bzw. 'P*x*:'. *x* ist hierbei die ein- bis zweistellige Nummer der gewünschten P-Taste. Der am Bildschirm angezeigte Eintrag aus dem Kommandogedächtnis ist mit ENTER bzw. ◄ ENTER zu senden. Die Eingabe wird damit in die angegebene P-Taste geladen. Die Ausführung der Eingabe als Kommando wird in diesem Fall unterdrückt.

Bei Angabe der Prefix 'P*x*#' wird der in die P-Taste geladene String mit ◄ ENTER abgeschlossen. Durch die Prefix 'P*x*:' wird kein ◄ ENTER generiert.

Beispiel:

Aus dem Kommandogedächtnis wird die Eingabe `STA 7813` am Bildschirm angezeigt. Durch Modifizieren der Eingabe nach `P12#STA 7813` und Absenden mit ENTER wird STA 7813 in die Taste P12 übertragen. Bei Betätigung von P12 wird der String automatisch mit Endemarke und ENTER abgesendet.

Ganzseitige Ausgabe des Kommandogedächtnisses



--◄ ENTER
--*string*◄ ENTER
--**string*◄ ENTER

Wird das Kommandogedächtnis durch Eingabe von '--' aktiviert, so wird ein ganzer Bildschirm (24 Zeilen) mit Eingaben des gewünschten Typs angezeigt. Die früheren Eingaben können sowohl sequentiell, als auch assoziativ angezeigt werden.

Die Aktivierung einer der angezeigten Eingaben erfolgt durch Modifizieren eines Zeichens im ausgewählten Kommando. Das Kommando wird auch ausgewählt, wenn ein Zeichen mit dem gleichen Inhalt überschrieben wird.

Mit SDF können auch mehrere BS2000-Kommandos aktiviert werden durch Modifizieren der entsprechenden Zeilen am Bildschirm. Die einzelnen Kommandos werden in der Reihenfolge ausgeführt, in der sie am Bildschirm angezeigt werden.

Hinweis:

Auch bei der seitenweisen Anzeige des Kommandogedächtnisses können ein oder mehrere Zeilen aus dem Gedächtnis in verschiedene, frei wählbare P-Tasten übertragen werden. Die Vorgehensweise ist dabei die gleiche wie oben beschrieben ("Inhalte aus Kommandogedächtnis in P-Tasten laden").

Rechnen und Konvertieren

Die Rechen- und Konvertierungsfunktion von CFS kann auch in Connections genutzt werden. Alle für den Rechen- und Konvertierungsmodul bestimmten Eingaben müssen mit der Zeichenfolge ^= beginnen.

Arithmetische und logische Operationen:

Es werden die vier Grundrechenarten + - * /, sowie %-Rechnung und die logischen Operationen .AND., .OR., .XOR. angeboten. Als Operanden können hexadezimale oder dezimale Zahlen angegeben werden. Die Rechenausdrücke sind in folgendem Format anzugeben:

^=*zahl* [*op zahl*] ...

zahl Dezimal oder Hexadezimalzahl.
Zahlen, die nur Ziffern 0 bis 9 enthalten, werden als Dezimalzahlen interpretiert. Enthält eine Zahl die Zeichen A bis F, so wird sie als Hexadezimalzahl interpretiert. Das Anhängen des Zeichens **H** oder **X** an eine Zahl in dezimaler Schreibweise (z.B. 10H) bewirkt, daß diese Zahl als Hexadezimalzahl interpretiert wird.

op Arithmetischer oder logischer Operator. Die folgenden Operatoren stehen zur Verfügung:

+	Addition
-	Subtraktion
*	Multiplikation
/	Division
%	%-Rechnung
.+.	Addition ohne Berücksichtigung des Vorzeichens (Adressrechnung)
.-.	Subtraktion ohne Berücksichtigung des Vorzeichens (Adressrechnung)
.AND.	logische UND-Verknüpfung
.OR.	logische ODER-Verknüpfung
.XOR.	logisches Exklusiv-ODER

Die Abarbeitung der einzelnen Operationen erfolgt rein linear. Beispiel: ^=10+10*2 ergibt als Ergebnis 40 und nicht 30, wie dies nach der Regel "Punkt vor Strich" zu erwarten wäre. Gleiches gilt für alle logischen Operationen.

Beispiele:

`^=16+80H`
In der Systemzeile wird als Ergebnis der Addition der Wert 144 in dezimaler und hexadezimaler Darstellung (90H) ausgegeben.

`^=7FFFFD45.+.FFF`
Als Ergebnis der logischen Addition wird der Wert 80000D44 ausgegeben. Die arithmetische Addition =7FFFFD45+FFF würde zu der Fehlermeldung 'result too big ...' führen, da die größtmögliche positive Hexadezimalzahl 7FFFFFFF überschritten wird (Festpunktüberlauf).

`^=1200%14+1200%85`
Der angegebene Ausdruck wird in folgender Reihenfolge berechnet:
14% von 1200=168, 168+1200=1368, 85% von 1368=1162.

`^=7C`
Für die angegebene Zahl wird die dezimale und hexadezimale Darstellung ausgegeben: 7C= 124 (7C)

`^=1.OR.2.OR.4.OR.8`
Die vier Hexadezimaloperanden werden mit einem logischen Oder verknüpft. Als Ergebnis wird der Hexadezimalwert F ausgegeben.

`^=AFFE.AND.5001H`
Die zwei Hexadezimaloperanden werden mit einem logischen Und verknüpft. Als Ergebnis wird der Hexadezimalwert 0H ausgegeben.

`^=7FFE.XOR.FF0`
Die zwei Hexadezimaloperanden werden mit einem logischen Exklusiv-Oder verknüpft. Als Ergebnis wird der Hexadezimalwert 700EH ausgegeben.

Character- und Hexadezimalkonvertierung:

In hexadezimaler Schreibweise angegebene Strings werden in Character-Darstellung konvertiert. In Character-Darstellung angegebene Strings werden in hexadezimale Darstellung konvertiert. Die zu konvertierenden Strings sind in folgendem Format anzugeben:

^=X'*string*' | ^=[C]'*string*'

Bei der Konvertierung in Character-Format werden nicht darstellbare Zeichen durch '?' angezeigt.

Beispiele:

`^=X'C1C442'`
In der Systemzeile wird für X'C1C442' die Character-Darstellung C'AD?' angezeigt. Im Characterformat nicht darstellbare Zeichen werden durch '?' angezeigt.

`^='AFFE'`
In der Systemzeile wird für C'AFFE' die hexadezimale Darstellung X'C1C6C6C5' angezeigt.

Parameter-Maske für DO-Prozeduren in Connections

Falls eine Connection am gleichen Hostrechner und unter der gleichen Benutzerkennung wie der CFS-Grundprozeß eröffnet wurde, so wird für DO-Kommandos folgender Zusatzservice angeboten:

Die Parameter der DO-Prozedur werden auf Vollständigkeit geprüft. Falls einem in der PROC-Anweisung definierten Parameter noch kein Wert zugewiesen wurde - entweder über Standard-Vorbelegung oder über das DO-Kommando selbst - , wird von CFS eine Parameter-Maske am Bildschirm ausgegeben. In der Parameter-Maske erscheinen alle in der PROC-Anweisung definierten Parameter zusammen mit den entsprechenden Werten. Die so angezeigten Parameter können für den aktuellen Prozedurlauf verändert bzw. ergänzt werden.

Die Parameter-Maske kann auch unvollständig ausgefüllt abgesendet werden. In diesem Fall erhalten die leeren Parameter keinen Wert zugewiesen und werden ggf. im BS2000 durch Prompting angefordert.

Zum kurzen Informieren über die in einer DO-Prozedur verwendeten Parameter kann ein DO-Kommando folgender Form eingegeben werden: DO *prozedur*,?

Hinweise:

Die Parameter-Maske für DO-Prozeduren kann unter besonderen Bedingungen auch dann ausgegeben werden, wenn der Nebenprozeß nicht unter der eigenen Benutzerkennung eröffnet wurde. Voraussetzung hierfür ist, daß die DO-Prozedur bzw. die diese DO-Prozedur enthaltende LMS-Bibliothek für den Grundprozeß zugreifbar ist.

Die Ausgabe einer Parameter-Maske ist nicht möglich, wenn die Prozedur als temporäre Datei gespeichert ist.

Um die PROC-Anweisung der DO-Prozedur zu analysieren, muß CFS diese DO-Prozedur im Lese-Modus eröffnen. Falls die Prozedur/LMS-Bibliothek mit einem READ-Passwort belegt ist, so wird dieses von CFS angefordert.

Die zuletzt gestartete BS2000 DO-Prozedur mit den zuletzt verwendeten Parametern wird aktiviert durch das Kommando DO ⊔ ◄ ENTER.

Durch das Kommando SP %*name* können die in der Parameter-Maske eingetragenen Parameter in einer Datei CFS.JRSAVE[.*jobname*] bzw. in einer mit /FILE *datei*, LINK=JRSAVE zugewiesenen Datei gespeichert werden. Die genaue Regel, nach der der Name der JRSAVE-Datei gebildet wird, ist auf Seite **564** bei CFS.JRSAVE beschrieben. Der Name der JRSAVE-Datei kann auch direkt im Anschluß an %*name* angegeben werden: SP %*name,jrsave*. %*name* steht für einen beliebigen ein- bis 44-stelligen Merkbegriff.

Ein DO-Kommando mit den unter %*name* gespeicherten Parametern kann wieder aktiviert werden durch das BS2000-Kommando DO %*name* bzw. DO %*name,jrsave*. Mehr zum Thema "Abspeichern von DO-Parametern" finden Sie auf Seite **403**.

Die automatische Prüfung der DO-Parameter und die Ausgabe der Parameter-Maske kann unterdrückt werden mit dem Kommando NDP (No Do-Parameters) oder indem das DO-Kommando mit einem führenden Schrägstrich eingegeben wird /DO *proc*.

TCHNG- oder /MODIFY-TERM-OPT in Connections

CFS führt für jede Connection getrennt eine eigene Überlaufsteuerung durch. Standardmäßig wird die Überlaufsteuerung des Grundprozeßes in die Connection übernommen. Innerhalb der Connection zu $DIALOG bzw. $CONSOLE kann jedoch durch Angabe des Kommandos TCHNG oder MODIFY-TERMINAL-OPT (beliebige SDF-Abkürzung möglich) für jede Verbindung eine individuelle Überlaufsteuerung festgelegt werden.

TCHNG oder MODIFY-TERM-OPT Kommandos müssen für den Connection-Handler sichtbar sein, d.h. sie müssen entweder über die Tastatur eingegeben werden oder als Ausgabe am Bildschirm erscheinen. Unsichtbare Kommandos sind z.B. solche, die in einer /PROC N Prozedur oder in einem Benutzerprogramm abgesetzt wurden. Unsichtbare Kommandos zur Überlaufsteuerung sind für die Connection ohne Wirkung.

Ein für den Connection-Handler sichtbares Kommando zur Überlaufsteuerung kann z.B. als BS2000-Kommando in einer /PROC C oder /PROC A Prozedur enthalten sein. Innerhalb einer /PROC N-Prozedur ist das Kommando WRITE-TEXT geeignet, um die Überlaufsteuerung für den Connection-Handler sichtbar zu machen.
Beispiel: `/WRITE-TEXT  'MODIFY-TERMINAL-OPT OVERFLOW=*NO'`

Bei Connections zu $CONSOLE kann das Kommando zur Veränderung der Überlaufsteuerung nach der Betätigung der K2-Taste ("please enter Console input (input/R/...)" in der gleichen Weise wie bei $DIALOG angegeben werden.

Mitschneiden von Dialogen (LOG/NLOG)

Innerhalb einer Connection kann der Benutzer das Mitschreiben (Logging) der Ein-/Ausgabe-Nachrichten durch CFS in eine Datei veranlassen. Der Beginn des Mitschnitts erfolgt durch das Kommando LOG. Es können sowohl BS2000-Dialoge, als auch UTM- und DCAM-Dialoge mitgeschnitten werden.

Dialogmitschnitt einschalten

LOG {*datei* | [*bibl*] ([*t*/] *element*) | STD | B | BR | C} [, *opt*]

datei — Name der Datei, in die die Ein-/Ausgaben geschrieben werden. Die LOG-Datei wird stets im CFS-Grundprozeß angelegt.

bibl ([*t*/] *element*) — Name der PLAM-Bibliothek und des Elements, in dem die aufgezeichneten Nachrichten gespeichert werden soll. Der Typ des Elements wird als S (Source) angenommen, falls nichts weiter angegeben wurde.

([*t*/] *element*) — Bei Nichtangabe des Bibliotheksnamens wird der Name der CFS-Userlib ergänzt. Für eine ausführliche Beschreibung der Userlib siehe Seite **541** und folgende.

STD Aufgrund der Eingabe STD wird von CFS ein Standardname für die LOG-Datei verwendet: CFS.LOG.*yymmdd. hhmmss.tsn*.

B Break. Der Vorgang des Mitschneidens wird unterbrochen bis zur Eingabe des Kommandos LOG C (LOG Continue, siehe unten).

BR wie LOG B (siehe oben). Vor der Suspendierung des Mitschnitts wird in die LOG-Datei ein Satz mit dem Inhalt 'STEP BREAK' eingefügt. Beim späteren Ablaufenlassen des Dialogs (RES) bewirkt der STEP BREAK-Satz, daß der RES-Vorgang bis zur Eingabe des Kommandos RES C (RES continue) ausgesetzt wird. Dies ermöglicht es dem Benutzer, mehrere Eingaben zu tätigen, die nicht in der LOG-Datei enthalten sind.

C Continue. Der unterbrochene Vorgang des Mitschneidens der Ein-/Ausgabenachrichten wird wieder aufgenommen. Zum Unterbrechen des Nachrichtenmitschnitts siehe oben (Parameter B/BR).

opt Über diesen Zusatzparameter kann die Art und der Umfang der aufzuzeichnenden Nachrichten bestimmt werden.

- I Input. Es werden nur die Eingaben aufgezeichnet (Standard).
- S Special. Es werden nur die Eingaben aufgezeichnet. (Unterschied zu I: siehe Hinweis 3).
- O Output. Es werden nur die Ausgaben aufgezeichnet.
- IO Es werden die Eingaben und Ausgaben aufgezeichnet.
- M preModify. Bei Bildschirmformaten werden die Inhalte aller Eingabefelder, auch solcher, die vom Benutzer nicht verändert wurden, aufgezeichnet.

Die M-Option kann, durch ein Komma getrennt, zusammen mit der I-, S-, bzw. IO-Option angegeben werden.

Dialogmitschnitt beenden

NLOG Die LOG-Datei wird geschlossen. Alle weiteren Eingaben werden nicht mehr mitgeschnitten.

Hinweise:

Das LOG-Kommando kann innerhalb einer Connection sowohl im Line-Modus, als auch im Maskenmodus eingegeben werden. Bei einer Bildschirmmaske ist das LOG-Kommando im ersten modifizierbaren Feld einzugeben. Eine Ausnahme bildet der EDT. Hier ist das LOG-Kommando in der 24-ten Zeile (Kommandozeile) einzugeben. In der gleichen Maske darf außerdem kein weiteres Feld modifiziert werden.

Bei Bildschirmmasken, in denen das erste modifizierbare Feld nicht zur Aufnahme des vollständigen LOG-Kommandos ausreicht, kann der Bildschirm vor Eingabe des LOG-Kommandos mit Shift Reset Bild gelöscht werden.

Beim Wiederholen eines mit der S-Option aufgezeichneten Dialogs können die generierten Eingaben am Bildschirm nicht dargestellt werden. Beim Wiederholen eines mit der I-Option aufgezeichneten Dialogs (Standardfall) werden die Eingaben und Ausgaben am Bildschirm dargestellt.
Aufgrund der S-Option werden in der LOG-Datei nur die reinen Eingaben und keine weiteren Hilfsinformationen aufgezeichnet.

Die M-Option (preModify) wirkt sich nur bei der Aufzeichnung von Masken-/ Formateingaben aus (SAW: "lese modifizierte Felder"). Die M-Option bewirkt, daß die Inhalte aller modifizierbaren Maskenfelder in der LOG-Datei protokolliert werden. Die M-Option bleibt ohne Wirkung bei Line-Mode Eingaben.

Beispiel: EDT-Sitzung im F-Modus. Wird eine EDT-Bildschirmseite mit der F2-Taste überschreibbar gemacht, so werden die Inhalte aller angezeigten Datenzeilen in der LOG-Datei festgehalten. Die Wirkung ist damit die gleiche, wie wenn alle Datenzeilen durch den Benutzer modifiziert worden wären.

Die Kommandos LOG und NLOG werden auch wirksam, falls sie in einer DO-Prozedur mit /WR-T 'LOG ...', bzw. /WR-T 'NLOG' als Kommentare am Bildschirm ausgegeben werden.

Der Systemverwalter kann über eine wahlweise Option CFS so generieren, daß das Fluchtsymbol ^ vor für den Connection-Handler von CFS bestimmten Kommandos angegeben werden muß. Das Ein-, bzw. Ausschalten des Loggings geschieht in diesem Fall durch ^LOG, bzw. ^NLOG. Ob die ^-Option in CFS aktiviert wurde, kann durch Aufrufen der CRD-Maske (Kommando CRD) erkannt werden.

Nach Einschalten des Loggings (Kommando LOG ...) werden die Eingaben STEP ON/ OFF/BREAK vom Connection-Handler intern als Kommandos für den späteren Restore-Vorgang interpretiert und als solche auch in der LOG-Datei gespeichert. Es erfolgt jedoch keine Weiterleitung der Eingaben an den Kommunikationspartner.

STEP ON Einschalten des Step-Modus. Beim späteren Wiederablaufenlassen des Dialogs (Kommando RES ...) wird bei dem aktuellen Dialogschritt der "Step-Modus" (siehe Seite 362) eingeschaltet. Der Step-Modus bleibt bis zu dem Dialogschritt eingeschaltet, bei dem das Kommando STEP OFF eingegeben wurde.

STEP OFF Ausschalten des Step-Modus. Beim Wiederablaufenlassen des Dialogs (Kommando RES ...) wird der Step-Modus ausgeschaltet. Die nachfolgenden Eingaben werden wie beim normalen Restore ohne Quittierung durch den Benutzer abgesendet.

STEP BREAK Zeitweilige Suspendierung des LOG-Vorgangs. Es können beliebig viele Eingaben getätigt werden, die nicht mitgeschnitten werden. Der LOG-Vorgang wird wieder aktiviert durch das Kommando LOG C (LOG continue).
Beim späteren Wiederablaufenlassen des Dialogs (Kommando RES ...) wird auf Benutzereingabe umgeschaltet und der RES-Vorgang wird unterbrochen. Es können beliebig viele Benutzereingaben getätigt werden. Der RES-Vorgang wird an der unterbrochenen Stelle fortgesetzt, sobald der Benutzer das Kommando RES C (RES continue) eingibt.

Die Eingaben STEP ON/OFF/BREAK können auch in Maskenformaten angegeben werden (im ersten beschreibbaren Feld). Da STEP ON/OFF/BREAK vom Connection-Handler nicht an den Partnerprozeß weitergeleitet wird, muß der Benutzer seine Echteingaben im anschließenden Dialogschritt eingeben. Die entsprechende Eingabeaufforderung wird wiederholt.

LOG-Dateien können als Elemente (Typ=S) in PLAM-Bibliotheken gespeichert werden. Das RES-Kommando ist entsprechend auch auf PLAM-Bibliothekselemente vom Typ S anwendbar (RES *bibl*(*element*)).

Ablaufenlassen von mitgeschnittenen Dialogen (RES)

Ein mit Hilfe des LOG-Kommandos mitgeschnittener Dialog kann innerhalb einer Connection jederzeit wieder zum Ablauf gebracht, d.h. mit identischen Eingaben wiederholt werden. Die Benutzereingaben werden dabei nicht von der Tastatur, sondern aus der LOG-Datei gelesen. Um einen mitgeschnittenen Dialog wieder ablaufen zu lassen, ist das Kommando RES (Restore) einzugeben.

Das RES-Kommando existiert in vier Varianten, die jeweils eine unterschiedliche Art des Dialogablaufs bewirken.

Restore: Batch-Modus

Im Batch-Modus läuft der zu wiederholende Dialog automatisch ab, ohne daß weitere Eingaben seitens des Benutzers notwendig sind.

RES [*datei* | *bibl* ([*t*/] *element*) | *element* | C] [, DELAY=*d*|ORIG]

datei Name der LOG-Datei, aus der die Dialogeingaben gelesen werden.
Die angegebene Datei muß vom CFS-Grundprozeß zugreifbar sein.

bibl ([t/] *element*) Name einer PLAM-Bibliothek und des entsprechenden Elements. Die durch das Logging erzeugte ISAM-Datei muß zuvor mit ADD in die Bibliothek aufgenommen werden. Der Typ des Elements wird als S (Source) angenommen, falls nichts weiter angegeben wurde. Um den Dialog ablaufen zu lassen, wird aus dem Bibliothekselement eine temporäre Datei erzeugt.

element Falls eine LOG-Datei mit dem im RES-Kommando angegebenen Namen im CFS-Grundprozeß nicht existiert, so wird von CFS unterstellt, daß ein gleichnamiges Element aus der CFS-Userlib gemeint ist. Bei der Abarbeitung des RES-Kommandos versucht CFS daher, das Element 'S/*element*' aus der CFS-Userlib zu selektieren und auszuführen. Für eine ausführliche Beschreibung der Userlib siehe Seite 541 und folgende.

C Continue. Ein durch STEP BREAK suspendierter RES-Vorgang wird an der unterbrochenen Stelle fortgesetzt. Das Anhalten des RES-Vorgangs kann während des Aufzeichnens (Logging) durch das Kommando LOG BR vordefiniert werden.

DELAY=*d* Verzögerungsfaktor. Die aus der LOG-Datei gelesenen Eingaben werden mit einer Verzögerung an den Kommunikationspartner weitergegeben. Diese Verzögerung erlaubt es, den Dialog langsamer ablaufen zu lassen, als dies im unverzögerten Modus der Fall wäre. Der am Bildschirm automatisch ablaufende Dialog kann damit in einer angenehmen Lesegeschwindigkeit vom Benutzer mitverfolgt werden.

n: 0 - 100 Verzögerung der Eingaben in Sekunden.
n: 100 - 999 Verzögerung der Eingaben in Millisekunden.

Standard: n=0. Die Eingaben werden ohne Verzögerung an den Kommunikationspartner weitergeleitet.

DELAY=ORIG Die aus der LOG-Datei gelesenen Eingaben werden mit der gleichen Verzögerung wie beim Mitschnitt (LOG-Kommando) an den Kommunikationspartner weitergegeben. Dadurch kann das originale Zeitverhalten bezüglich der Eingabe (Denkpausen) simuliert werden.



Durch Drücken der K2-Taste kann aus dem Batch-Modus des Restore in den Step-Modus (siehe unten) verzweigt werden. Durch Angabe eines Verzögerungsfaktors wird der Dialogablauf so verlangsamt, daß die K2-Taste sicher zwischen zwei Dialogschritten betätigt werden kann.

Beispiel:

```
RES CFS.LOG,DELAY=2
```

Jede Eingabe aus der LOG-Datei CFS.LOG wird mit 2 Sekunden Verzögerung an den Kommunikationspartner gesendet.

Hinweise:

Das RES-Kommando kann sowohl im Line-Modus, als auch im Maskenmodus eingegeben werden. In einer Bildschirmmaske ist das RES-Kommando im ersten modifizierbaren Feld einzugeben. Eine Ausnahme hiervon bildet der EDT. Hier ist das RES-Kommando in der 24-ten Zeile (Kommandozeile) des EDT einzugeben. In der gleichen Maske darf außerdem kein weiteres Feld modifiziert werden.

Bei Bildschirmmasken, in denen das erste modifizierbare Feld nicht zur Aufnahme des RES Kommandos ausreicht, kann Shift Reset Bild verwendet werden, um den Bildschirm zu löschen und in Zeile 1, Spalte 1 das RES-Kommando einzugeben. Die Anwendung von Shift Reset Bild ist in Connections jederzeit möglich, insbesondere bei Formaten, die das Löschen des Bildschirmspeichers normalerweise sperren.

Das Kommando RES wird auch wirksam, falls es in einer DO-Prozedur mit /WR-T 'RES ...' als Kommentar am Bildschirm ausgegeben wird.

Die Eingabe 'RES ' (RES + Blank, jedoch ohne Angabe eines Dateinamens) bewirkt, daß die im letzten LOG-/RES-Kommando angegebene Datei für den Restore-Vorgang verwendet wird.

Ein mit Restore wieder zum Ablauf gebrachter Dialog kann durch das Hardcopy-Kommando (HC ...) bildschirmgerecht in einer druckaufbereiteten Datei protokolliert werden. Dies gilt auch im Phantom-Modus (siehe unten) des Restore Dialogs. Zum gleichen Thema siehe auch Seite **368** "Hardcopy-Modus in Connections".

Bei SETSW ON=4 im Hauptprozeß von CFS wird die Anzeige der durch den RES-Vorgang erzeugten Eingaben unterdrückt.

Ein Dialog kann durch Restore nur dann wiederholt werden, wenn dieser beim Logging mit den Optionen I (Input [= Standard]), oder IO (Input + Output) protokolliert wurde.

Der Systemverwalter kann über eine wahlweise Option CFS so generieren, daß das Fluchtsymbol ^ vor für den Connection-Handler von CFS bestimmten Kommandos angegeben werden muß. Das Einschalten des Restore-Vorgangs geschieht in diesem Fall durch ^RES. Ob die ^-Option in CFS aktiviert wurde, kann durch Aufrufen der CRD-Maske (Kommando CRD) erkannt werden.

Restore: Step-Modus

Im Step-Modus des Restore werden die aus der LOG-Datei gelesenen Eingaben erst nach Bestätigung durch den Benutzer an den Kommunikationspartner gesendet. Die vorgeschlagenen Eingaben können verändert werden, so daß sich der aktuelle Dialog unter Umständen vom mitgeschnittenen Dialog unterscheidet.

RES [*datei* | *bibl* ([*tl*] *element*) | *element* | C] **, S** [, DELAY=*d*]

S — Step-Modus. Dieser Parameter bewirkt, daß der Restore-Vorgang im Step-Modus abläuft. Im Step-Modus wird jeweils eine Eingabe aus der Datei gelesen und am Bildschirm angezeigt. Die angezeigte Eingabe kann vom Benutzer noch abgeändert und danach mit ENTER abgesendet werden.

DELAY=*d* — Verzögerungsfaktor.
Im Step-Modus des Restore wird durch Angabe eines Verzögerungsfaktors erreicht, daß Line-Mode Ausgabenachrichten nach einer gewissen Wartezeit durch die nächste Eingabe, die z.B. im Format-Modus erfolgen kann, überschrieben werden.

d: 0 - 100 Verzögerung der Eingaben in Sekunden.
d: 100 - 999 Verzögerung der Eingaben in Millisekunden.

K- und F-Tasten im Step-Modus:

K1 — Durch Betätigung der K1-Taste wird der Restore-Vorgang vorzeitig abgebrochen.

F2 — Das Absenden des veränderten oder unveränderten Eingabevorschlags mit der F2-Taste bewirkt, daß der aktuelle Eingabevorschlag beim nächsten Dialogschritt wieder angezeigt wird. Es wird nicht der nächste Satz aus der LOG-Datei gelesen. Es kann damit eine zusätzliche, nicht in der LOG-Datei aufgezeichnete Eingabe in den Dialog eingefügt werden.

F3 — Das Absenden eines unveränderten oder veränderten Eingabevorschlags mit der F3-Taste bewirkt den Übergang in den "Batch-Modus" des Restore.

Restore: Compare-Modus (Regressionstest)

Im Compare-Modus des Restore werden die vom Kommunikationspartner empfangenen Nachrichten mit den in der LOG-Datei gespeicherten verglichen.

RES [*datei* | *bibl* ([*tl*] *element*) | *element*] **, C** [, A] [, HC=*datei* [, *hc-params*]]

C — Compare-Modus. Die von der Connection empfangenen Ausgabe-Nachrichten werden mit den in der LOG-Datei gespeicherten Nachrichten verglichen. Die LOG-Datei muß mit **LOG** *datei*,**IO** erstellt worden sein.

A — Automatische Regression. Der Compare-Dialog läuft automatisch, d.h. ohne Benutzereingaben bis zum Ende. In Verbindung mit dem Parameter A sollte auch der nachfolgende Parameter HC= angegeben werden.
Standard: Kein automatischer Regressionstest. Der Benutzer muß jede Differenz im Ausgabeverhalten des zu testenden Programms bestätigen.

HC=*datei* [, *params*]

Hardcopy aller Ausgaben, die sich vom mitgeschnittenen Masterdialog unterscheiden. In der angegebenen Datei werden die abweichenden Bildschirmmasken in der Form NEW und DIF (siehe unten) protokolliert. Für *params* können alle optionalen Parameter des HC-Kommandos, wie z.B. T (Time-Stamps) oder L (Laseraufbereitung) angegeben werden.

Ablauf des Compare-Modus:

Der Restore-Vorgang läuft automatisch, d.h. in der gleichen Weise wie der normale Batch-Restore (Variante 1) ab.

Werden beim Vergleich der in der LOG-Datei enthaltenen Ausgabenachrichten mit den aktuellen Ausgabenachrichten Unterschiede festgestellt, so wird der Benutzer in der Systemzeile des Bildschirms (Zeile 25) darauf hingewiesen. Der Restore-Vorgang wird, ähnlich wie im Step-Modus, angehalten.

Folgende K-/F-Tasten dienen der Steuerung des weiteren Ablaufs:



Der Benutzer kann durch Betätigen der K3-Taste zwischen der Anzeige der aktuellen und der in der LOG-Datei enthaltenen Ausgabe-Nachrichten wechseln. In der Systemzeile werden die entsprechenden Nachrichten durch NEW und OLD gekennzeichnet. Durch nochmaliges Drücken der K3-Taste wird eine Differenz-Nachricht angezeigt, bei der alle Stellen von NEW auf Spaces gesetzt sind, mit Ausnahme derer, bei denen sich NEW von OLD unterscheidet. Diese Nachricht ist in der Systemzeile mit DIF gekennzeichnet.



Durch Betätigung der K1-Taste während der Anzeige einer NEW/OLD/DIF-Nachricht wird der gesamte Restore-Vorgang abgebrochen.

ENTER
Wird die NEW/OLD/DIF-Nachricht mit der ENTER-Taste abgesendet, so bewirkt dies die Fortsetzung des Restore-Vorgangs.



Über die F2-Taste gelangt der Benutzer in ein Menü für zusätzliche Test-Optionen.

Auswahlmenü: Test-Optionen

SN
Skip New: Die nächste Ausgabe des Kommunikationspartners wird mit der gegenwärtig im Zugriff befindlichen LOG-Ausgabenachricht verglichen. Diese Option ist gleichbedeutend mit dem Herausnehmen der Ausgabe des Kommunikationspartners aus dem zu vergleichenden Nachrichtenstrom.

SO
Skip Old: Die nächste in der LOG-Datei enthaltene Ausgabe wird mit der aktuellen Ausgabe des Kommunikationspartners verglichen. Diese Option ist gleichbedeutend mit dem Herausnehmen der in der LOG-Datei enthaltenen Ausgabenachricht aus dem zu vergleichenden Nachrichtenstrom.

E
Edit: Hiermit gelangt der Benutzer in den Compare-Editor (siehe unten). Im Compare-Editor kann die in der LOG-Datei festgehaltene Ausgabenachricht editiert werden. Insbesondere können bestimmte Felder durch Überschreiben mit '#' vom Vergleich ausgeschlossen (maskiert) werden. Nach Verlassen des Editors (Drücken der ENTER-Taste) wird der letzte Vergleichsschritt mit der soeben editierten LOG Ausgabe-Nachricht wiederholt.

R
Return: Verlassen des Menüs "Test-Optionen".

Hinweis:

Der automatische Modus des Regressionstests ist nicht an eine Datensichtstation gebunden und kann auch im Batch durchgeführt werden. Im Hardcopy-Protokoll werden die Masken in diesem Fall jedoch nicht bildschirmgerecht aufbereitet.

Beispiel:

Kommandofolge zum Aufruf des automatischen Regressionstests im Batch:

```
/SYSFILE SYSDTA=(SYSCMD)
/SETSW ON=(1,4)
/EXEC CFS
*002 FILE=NO
OC1/LOGON ......
RES datei,C,A,HC=CFS.HARDCOPY,O
LOGOFF NOSPOOL
END
```

Der Compare-Editor

Der Compare-Editor ist eine Einrichtung zum Nachbearbeiten des für den Regressionstest mitgeschnittenen Musterdialogs. Es können bestimmte Nachrichten oder Teile von Nachrichten (z.B. Datum und Uhrzeit in Bildschirmmasken) vom späteren Vergleich während des Restore ausgeschlossen werden.

Der Compare-Editor wird durch das folgende Restore-Kommando aufgerufen:

RES *datei* **, CE** [, *n*]

CE — Der Compare-Editor wird zur Bearbeitung der in der LOG-Datei gespeicherten Ausgabenachrichten aktiviert. Dabei werden alle Stellen der Ausgabe, insbesondere auch die geschützten Felder, dem Benutzer zum Modifizieren freigegeben.

Es können Teile der Ausgabenachrichten vom späteren Vergleich ausgeschlossen werden. Dies geschieht, indem die gewünschten Stellen der Nachricht mit dem Zeichen '#' überschrieben (maskiert) werden.

Soll eine Ausgabe als Ganzes vom Vergleich ausgeschlossen werden, so ist am Bildschirm in Zeile 1, Spalte 1 das Zeichen ';' einzutragen.

Der Compare-Editiervorgang kann durch Betätigung der K1-Taste jederzeit abgebrochen werden.

Die in der LOG-Datei gespeicherten Ausgabenachrichten werden innerhalb des Compare-Editor durch eine laufende Nummer (001, 002, ...) identifiziert.

n — Id-Nummer für Nachricht. *n*: 0 - 999
Die Angabe einer Id-Nummer im RES-Kommando bewirkt, daß der Compare-Editor die Anzeige direkt bei der Ausgabenachricht mit der angegebenen Nummer beginnt.

Hinweise:

Der CAP-/LOW-Modus der CFS-Grundebene wirkt auch im Compare-Editor: Bei CAP (=Standard) werden eingegebene Kleinbuchstaben in die entsprechenden Großbuchstaben umgesetzt. Im LOW-Modus werden eingegebene Kleinbuchstaben nicht umgesetzt.

Bei Bildschirm-Formaten ist zu beachten, daß der Compare-Editor die Nachrichten im allgemeinen nicht zeilen- und spaltengerecht darstellt. Auch die Tabulatortaste zum Positionieren zwischen den einzelnen Maskenfeldern kann im Compare-Editor Modus nicht verwendet werden.

Der Compare-Editor eignet sich auch zum Forcieren von Unterschieden bei einem späteren Compare-Lauf. (Inhaltliche Modifikation der Ausgaben).

Restore: Phantom-Modus

Der in der LOG-Datei mitgeschnittene Dialog wird am Bildschirm originalgetreu simuliert. Es findet dabei kein Nachrichtenverkehr mit der Anwendung statt.

RES [*datei* | *bibl* ([*tl*] *element*) | *element*] **, P** [, S] [, DELAY=*d*]

P Phantom-Modus. Die in d%r LOG-Datei enthaltenen Eingaben werden am Bildschirm dargestellt, jedoch nicht an die Connection weitergereicht. Die als Reaktion auf die Eingaben empfangenen Ausgabenachrichten werden im Phantom-Modus ebenfalls aus der LOG-Datei gelesen und am Bildschirm dargestellt.

Der Phantom-Modus des Restore ist nur möglich, falls die benutzte LOG-Datei mit **LOG** *datei*,**IO** erstellt wurde.

S Der simulierte Dialog läuft schrittweise, d.h. im Step-Modus ab. Der Benutzer muß jede am Bildschirm angezeigte Eingabe mit der ENTER-Taste quittieren.

Im Step-Modus des Phantom-Restore lassen sich **Dialogschritte überspringen** und **wiederholen**. Um während des Phantom-Dialogs einige Schritte nach vorne oder nach hinten zu springen, ist der gesamte Bildschirm zu löschen (Taste Shift Reset Bild). Im gelöschten Bildschirm kann nun links oben die Anzahl der Vorwärts-/Rückwärtsschritte eingegeben werden: +*n*/-*n*/++/--◄ (*n*: 1 - 99). Die Eingabe muß mit der **F1**-Taste gesendet werden.

DELAY=*d* Verzögerung zwischen den Ausgabeschritten.

n:	0 - 100	Verzögerung in Sekunden.
n:	100 - 999	Verzögerung in Millisekunden.

Standard: Das Zeitverhalten des Originaldialogs wird simuliert. Wartezeiten von mehr als einer Minute werden dabei auf 60 Sekunden reduziert. Für weitere Informationen siehe Hinweise.

Beispiele:

```
RES CFS.LOG,P,DELAY=1
```
Nachrichten werden mit einer Sekunde Verzögerung am Bildschirm ausgegeben.

```
RES CFS.LOG,P,DELAY=0
```
Schnellstmögliche Ausgabe der Nachrichten am Bildschirm.

Hinweise:

Falls weder der Step-Modus noch ein Zeitverzögerungsfaktor angegeben wurden (RES ...,P), so wird das Zeitverhalten des Ursprungsdialogs simuliert. Die zeitliche Aufeinanderfolge der einzelnen Ein-/Ausgabeschritte ist dann die gleiche wie im Ursprungsdialog. Wartezeiten von mehr als einer Minute werden auf 60 Sekunden reduziert. Falls der Benutzer das Intervall zwischen 2 Dialogschritten nicht abwarten möchte, so kann er die Wartezeit durch Drücken der K2-Taste beenden.

Da Dialoge mit UTM-/DCAM-Anwendungen im Phantom-Modus des Restore nur simuliert werden, ist es nicht erforderlich, daß vor Ausführung des Restore eine Connection zu der entsprechenden Anwendung aufgebaut ist. Es kann z.B. ein UTM-Dialog in einer $DIALOG-Connection simuliert werden.

Ein mit Restore wieder zum Ablauf gebrachter Dialog kann mit Hilfe des Hardcopy-Kommandos (HC ...) bildschirmgerecht in einer druckaufbereiteten Datei protokolliert werden. Siehe hierzu Seite 368 "Hardcopy-Modus in Connections". Das bildschirmgerechte Protokollieren des Dialogs ist auch im Phantom-Modus des Restore möglich.



Falls der Phantom-Dialog im Step-Modus abläuft, kann der Benutzer durch Löschen des Bildschirms und Eintragen von +*n*/-*n*/++/-- in Zeile 1, Spalte 1 die Anzahl von Dialogschritten angeben, um die der Dialog vor oder zurück gesetzt werden soll.

+*n* Dialog um *n* Schritte vorsetzen.

++ Verzweigen zum letzten mitgeschnittenen Dialogschritt.

-*n* Dialog um *n* Schritte zurücksetzen.

-- zum Anfang des Dialogmitschnitts zurückpositionieren.

Die Eingaben müssen mit der F1-Taste abgesendet werden.



Falls der Phantom-Dialog nicht im Step-Modus abläuft, kann durch Betätigen der K2-Taste die Wartezeit zwischen zwei aufeinanderfolgenden Dialogschritten abgekürzt werden. Durch mehrmaliges, schnell aufeinanderfolgendes Betätigen der K2-Taste wird dem Benutzer die Möglichkeit gegeben, den Phantom-Dialog abzubrechen (CFS-Meldung: "Abort Restore ? (Y/N)").

Verknüpfen von Logging und Restore

Ein in der Praxis gelegentlich auftretendes Problem besteht darin, daß der Benutzer einen Dialogmitschnitt nachträglich modifizieren möchte. Eine Lösungsmöglichkeit dieses Problems besteht darin, den zu modifizierenden Dialog unter Zuhilfenahme des ursprünglichen Dialogmitschnitts neu aufzuzeichnen. Folgende Vorgehensweise ist dazu notwendig:

Zuerst muß mit LOG *datei2* eine neue LOG-Datei eröffnet werden.

Als zweiter Schritt ist der "Master-Dialog" im Step-Modus zum Ablauf zu bringen: RES *datei1* ,S.

Es werden nun alle in der ursprünglichen LOG-Datei enthaltenen Eingaben in die neue LOG-Datei übernommen, wobei der Benutzer die Möglichkeit hat, den Master-Dialog durch Modifikationen der vorgeschlagenen Eingaben zu ändern.

Das Beenden des Loggings durch das Kommando NLOG sollte am Ende des Dialogs nicht vergessen werden.

HELP-Informationen zum Thema Connections ausgeben

?CON Es werden die HELP-Informationen zum Thema "Connection-Handling" ausgegeben. Diese Zeichenfolge kann auch in Connections sowohl im Line- als auch im Format-Modus (im ersten Feld der Maske) eingegeben werden.

Übersicht über alle eröffneten Connections (Connection-Display)

CD Connection Display. Mit diesem Kommando kann man sich eine Übersicht der im Moment eröffneten Connections verschaffen. Am Terminal erscheinen die gerade eröffneten Nebenprozesse mit TSN und dem zuletzt eingegebenen Kommando.

Connection-Hardcopy Taste

In der Grundebene von CFS kann mit der F3-Taste ein Hardcopy für einzelne Masken angefordert werden. In Connections kann die F3-Taste durch eine Anwender-Funktion belegt sein. Aus diesem Grunde kann der Benutzer für Connections eine beliebige F- bzw. K-Taste als Äquivalent zur F3-Taste der CFS-Grundebene festlegen. Dies geschieht mit dem Kommando CH.

CH *t-bez* Connection Hardcopy. Das CH-Kommando ist im Kommandofeld in der Grundebene von CFS einzugeben. Die im CH-Kommando definierte Hardcopy-Taste gilt in allen Connections.

t-bez Tastenbezeichnung: K1, K3, ... ,K14 / F1, ... ,F20

Die mit dem CH-Kommandos definierte Hardcopy-Taste hat in Connections die gleiche Funktion wie die F3-Taste in der Grundebene von CFS. Eine mit der in CH festgelegten F-/K-Taste abgeschickte Maske wird in der Hardcopy-Datei protokolliert. Die getätigten Eingaben werden vom Connection-Handler jedoch nicht an den Kommunikationspartner weitergeleitet.

Die mit der Connection Hardcopy-Taste abgesendeten Maskeneingaben müssen nach der Protokollierung noch einmal eingegeben und mit ENTER gesendet werden.

Hardcopy-Modus in Connections

Befindet sich der Benutzer in einem Nebenprozeß von CFS, so kann er mit dem HC-Kommando das maskengerechte Mitschneiden des Dialogs in einer druckaufbereiteten Datei einschalten (Hardcopy-Modus).

HC [*datei*] [, L] [, B] [, T] [, E|O] [, A|F] [, G|I] [, S] [, RS] [, SMO]

Alle Optionen des HC-Kommandos sind wahlweise. Werden sie nicht angegeben, so gelten die entsprechenden Standardwerte.

datei Name der Datei, in die der Dialog protokolliert werden soll. Die Hardcopy-Datei wird stets im Grundprozeß von CFS erzeugt. Es können in jeder Connection eigene Hardcopy-Dateien angelegt werden.

Existiert die angegebene Hardcopy-Datei bereits und wurde keine der Optionen O/ E angegeben (siehe unten), so wird der Benutzer gefragt, ob die bestehende Hardcopy-Datei erweitert oder überschrieben werden soll.

Falls der Hardcopy-Vorgang in der Connection zeitweilig ausgeschaltet wurde (NHC) und mit HC später wieder eingeschaltet wird, so wird die zuletzt eröffnete Hardcopy-Datei erweitert.

Wird beim ersten HC-Kommando innerhalb einer Connection kein Dateiname angegeben, so wird eine Hardcopy-Datei mit dem Standardnamen CFS.HARDCOPY. *yymmdd.hhmm.tsn* angelegt.

L Laserdruck-Aufbereitung: Die Hardcopy-Datei wird im laserausdruckbaren Format (ND-/HPFILE) erzeugt. Helle/halbhelle, geschützte/ungeschützte Felder werden somit auch im Druckbild entsprechend dargestellt.

Die mit der L-Option erzeugte Hardcopy-Datei kann mit dem BS2000 Kommando /DO CFS.DO.PRINT,*datei* ausgedruckt werden. Die DO-Prozedur CFS.DO.PRINT, sowie die zugehörige CFS.ND-/HPFILE müssen unter TSOS zur Verfügung stehen und shareable sein. Der Systemverwalter muß die PROC-Anweisung dieser Prozedur evtl. noch bezüglich der verwendeten HP-/NDFILE und der Zeichensätze anpassen.

Hinweise hierzu sind in der Prozedurdatei enthalten. Standard: L-Option ausgeschaltet, d.h. die Hardcopy-Datei wird ohne besondere Aufbereitung für Laserdrucker erzeugt.

B — Before-Images: Bei Bildschirmmasken wird zusätzlich zur standardmäßigen Protokollierung nach der Benutzereingabe auch die Maske im Zustand vor der Eingabe protokolliert. Standard: B-Option ausgeschaltet (keine Before-Images).

T — Time-Stamps: Die protokollierten Ein-/Ausgaben werden zusammen mit Datum und Uhrzeit in der Hardcopy-Datei abgelegt.
Standard: T-Option ausgeschaltet (kein Time-Stamping).

E | O — Extend/Overwrite:

E — Falls die angegebene Hardcopy-Datei bereits existiert, wird sie mit Open=Extend eröffnet.

O — Die angegebene Hardcopy-Datei wird im Output-Modus eröffnet, d.h. der bisherige Inhalt wird überschrieben.

Standard: Falls eine Hardcopy-Datei gleichen Namens bereits existiert, wird der Benutzer gefragt, ob er die bestehende Datei überschreiben oder erweitern möchte.

A | F — 1) Mit Laseraufbereitung (**L-Option eingeschaltet**):

A — Die Anzeigesteuerzeichen der Bildschirmmasken werden auf hell und halbhell untersucht. Halbhelle Felder werden im grau hinterlegten Zeichensatz dargestellt.

F — Die Feldbehandlungszeichen der Bildschirmmasken werden auf geschützte und ungeschützte Felder untersucht. Geschützte Felder werden im grau hinterlegten Zeichensatz dargestellt.

Standard: A-Option (Anzeigesteuerzeichen werden ausgewertet).

2) Ohne Laseraufbereitung (**L-Option nicht eingeschaltet**):

F — In der Hardcopy-Datei werden die IS2-Folgen der Bildschirmmasken mit abgelegt. Dies gibt dem Benutzer die Möglichkeit, die IS2-Folgen selbst in geeignete Umschaltzeichen für besondere Zeichensätze des Laserdruckers umzuwandeln.

A — In der Hardcopy-Datei werden die IS3-Folgen der Bildschirmmasken mit abgelegt. Dies gibt dem Benutzer die Möglichkeit, die IS3-Folgen selbst in geeignete Umschaltzeichen für besondere Zeichensätze des Laserdruckers umzuwandeln.
Standard: A-/F-Option ausgeschaltet (IS2-/IS3-Folgen werden nicht in die Hardcopy-Datei übernommen).

Die F-/A-Option können auch zusammen angegeben werden. In der Hardcopy-Datei erscheinen dann sowohl die IS2-, als auch die IS3-Folgen.

G | I — Nur bei eingeschalteter L-Option: German/International

G — Deutsche Umlaute werden im Laserausdruck als solche dargestellt. Dies erfolgt unabhängig von der Darstellung am Bildschirm.

I — Deutschen Umlaute werden gemäß der internationalen Zeichendarstellung in der Hardcopy-Datei als eckige/geschweifte Klammern dargestellt.
Standard: G-Option (Darstellung der deutschen Umlaute).

S — Screen: Nach Eingabe des HC-Kommandos im Line-Modus werden normalerweise alle Ausgaben und Eingaben in der Hardcopy-Datei festgehalten. Der Inhalt des Bildschirms zum Zeitpunkt des HC-Kommandos wird dabei nicht erfaßt. Mit der S-Option

wird nach dem Einschalten des Hardcopy-Modus der gesamte, zu diesem Zeitpunkt sichtbare Bildschirminhalt in der Hardcopy-Datei festgehalten.
Die S-Option ist nur sinnvoll bei Eingabe des HC-Kommandos im Line-Modus.
Standard: S-Option nicht eingeschaltet.

RS — Reduced Size: Dieser Modus bewirkt beim Hardcopy von Bildschirmmasken, daß die Umrahmung der Masken unterdrückt wird. Außerdem werden in den Ausgabesätzen alle rechts stehenden Blanks gelöscht. Damit wird eine deutliche Reduzierung der Dateigröße besonders bei umfangreichen Hardcopy-Vorgängen erreicht.

SMO — Single Mask Output: Bei mehreren aufeinanderfolgenden Masken wird nur die letzte in der Hardcopy-Datei aufgezeichnet. D.h. nach jeder Maskenaufzeichnung wird die Hardcopy-Datei automatisch geschlossen und vor der nächsten Maske mit Open Output geöffnet.

Beenden des Hardcopy-Modus

NHC — No HardCopy. Der Hardcopy-Modus wird wieder ausgeschaltet. Die Hardcopy-Datei wird ebenfalls geschlossen, wenn die Connection verloren geht (Disconnect/ Connection Lost) oder durch LOGOFF/ KDCOFF beendet wird.

Hinweise:

Das HC-Kommando kann sowohl im Line-Modus als auch im Masken-Modus eingegeben werden. In einer Bildschirmmaske ist das HC-Kommando im ersten modifizierbaren Feld einzutragen. Eine Ausnahme hiervon bildet der EDT. Hier ist das HC-Kommando in der 24-ten Zeile (Kommandozeile) einzugeben. In der gleichen Maske darf außerdem kein weiteres Feld modifiziert werden. Dasselbe gilt für das NHC-Kommando.

Bei Bildschirmmasken, in denen das erste modifizierbare Feld nicht zur Aufnahme des HC-Kommandos ausreicht, kann mit Shift Reset der Bildschirm gelöscht und in Zeile 1, Spalte 1 das gewünschte HC-Kommando eingegeben werden.

Die Kommandos HC und NHC werden auch wirksam, falls sie in einer DO-Prozedur mit /WR-T 'HC ...', bzw. /WR-T 'NHC' am Bildschirm ausgegeben werden.

Der Systemverwalter kann über eine wahlweise Option CFS so generieren, daß das Fluchtsymbol ^ vor für den Connection-Handler von CFS bestimmten Kommandos angegeben werden muß. Das Ein-, bzw. Ausschalten des Hardcopy-Modus erfolgt in diesem Fall durch ^HC, bzw. ^NHC. Ob die ^-Option in CFS aktiviert wurde, kann durch Aufrufen der CRD-Maske (Kommando CRD) erkannt werden.

Menue-Maske für Connections aufrufen

MENUE [*datei*] — Es wird in das Menue-System für Connections verzweigt.
Weitere Informationen über die Menue-Maske finden Sie auf Seite 379 und folgende (Menue-System für Connections).

Unterdrückung der Ausgaben

Der Connection-Handler von CFS bietet die Möglichkeit, die Ausgabe aller in einem Nebenprozeß durch ein Benutzerprogramm oder durch das System produzierten Meldungen zu unterdrücken:

O OFF [,*datei*] Alle Ausgaben in der aktuellen Connection werden ab sofort am Bildschirm nicht mehr dargestellt. Die zu einer Eingabeaufforderung gehörige Nachricht (z.B. '/') ist dadurch nicht betroffen und wird am Bildschirm angezeigt.

datei Die optionale Angabe eines Dateinamens hat zur Folge, daß die unterdrückten Ausgaben in die angegebene Datei geschrieben werden.

O ON Alle Ausgabenachrichten für die Connection werden wieder angezeigt.

Hinweise:

Ein Anwendungsgebiet für das Kommando O OFF sind Programmtests mit AID wobei für einen bestimmten Haltepunkt ein Durchlaufzähler gesetzt wird. Der Haltepunkt kann sehr oft durchlaufen werden. Bei jedem Durchlauf wird von IDA eine Meldung am Bildschirm ausgegeben. Diese Bildschirmausgaben beanspruchen relativ viel Zeit und verzögern den Programm-Ablauf.

Durch die Eingabe O OFF kann der Durchlauf des zu testenden Programms insofern beschleunigt werden, als die von AID produzierten Bildschirmausgaben entfallen.

Die Kommandos O OFF und O ON werden auch wirksam, falls sie in einer DO-Prozedur mit /WR-T 'O OFF ...', bzw. /WR-T 'O ON' als Kommentare am Bildschirm ausgegeben werden.

Der Systemverwalter kann über eine wahlweise Option CFS so generieren, daß vor Kommandos, die für den Connection-Handler bestimmt sind, das Fluchtsymbol ^ angegeben werden muß. Das Unterdrücken und wieder Einschalten der Anzeige der Ausgabenachrichten geschieht in diesem Fall durch die Kommandos ^O OFF, bzw. ^O ON. Ob die ^-Option in CFS aktiviert wurde, kann durch Aufrufen der CRD-Maske (Kommando CRD) erkannt werden.

Asynchroner Modus für Connections

AC*n* Asynchron modus for Connection *n*.
Wenn der Benutzer durch Betätigen der K2-Taste die Connection verläßt, werden normalerweise alle von dieser Connection ausgegebenen Meldungen zwischengespeichert. Beim nächsten CC*n* (Continue in Connection) werden diese Meldungen am Bildschirm ausgegeben.

Dieser Standardmodus kann durch das Kommando AC*n* in der Weise modifiziert werden, daß nach Verlassen der Connection *n* alle dort eingehenden Nachrichten sofort am Bildschirm ausgegeben und nicht von CFS zurückgehalten werden bis der Benutzer wieder in den Nebenprozeß verzweigt. Dabei wird den von der Connection *n* produzierten Meldungen der Identifier (*n*:) vorangestellt. Damit können asynchrone Meldungen von verschiedenen Connections voneinander unterschieden werden. Der Asynchrone Modus kann für jede Connection getrennt ein- und ausgeschaltet werden (AC*n* / NAC*n*).

NAC*n* No Asynchron mode für Connection n.
Wiederherstellung des Normalmodus für Connection *n* (Zwischenspeicherung der asynchron eingehenden Meldungen bis zum nächsten CC*n*).
Standard: NAC-Modus für jede Connection.

Hinweis:

Asynchron-Modus und WAIT-Kommando:
Der Benutzer hat in mehreren Connections Consolanwendungen zu verschiedenen Host-Rechnern eröffnet. Alle Consol-Connections wurden durch AC-Kommandos in den Asynchron-Modus versetzt. Eine einfache Möglichkeit, die Meldungen aller Konsolen gesammelt an einem Bildschirm zu betrachten, ergibt sich über das WAIT-Kommando.

Automatischer / nicht automatischer Connection-Display

ACD | NACD Automatic Connection Display / No Automatic Connection Display.

ACD Bei Betätigung der K2-Taste erfolgt die Anzeige aller eröffneten Connections.

NACD Kein automatischer Connection Display bei Betätigung der K2-Taste.

Standard: ACD bei SETSW OFF=(4)
NACD bei SETSW ON=(4)

Connection-Hardcopy Taste

CH *t-bez* Connection Hardcopy.
In der Grundebene von CFS kann mit der F3-Taste ein Hardcopy für einzelne Masken angefordert werden. In Connections kann die F3-Taste durch eine Anwender-Funktion belegt sein. Aus diesem Grunde kann der Benutzer für Connections eine beliebige F- bzw. K-Taste als Äquivalent zur F3-Taste der CFS-Grundebene festlegen. Dies geschieht mit dem Kommando CH.

t-bez Tastenbezeichnung: K1, K3, ... ,K14 / F1, ... ,F20.

Standard: NONE (keine Tastenbelegung).

Hinweise:

Beim Hardcopy einer Maske durch Betätigen der durch das Kommando CH definierten Datenübertragungstaste gehen die Eingaben verloren und müssen noch einmal eingegeben und mit ENTER gesendet werden.

Es ist zu beachten, daß bei Hardcopy mit der durch CH definierten Datenübertragungstaste jede Einzelmaske im ausgedruckten Listing auf einer neuen Seite erscheint. Im normalen Hardcopy-Modus (HC-Kommando) werden nach Möglichkeit mehrere Masken auf einer Druckseite untergebracht.

BS2000-Prompt in Connections ändern

CP '*param*' In CFS-Connections wird als Eingabeaufforderung für BS2000-Kommandos eine frei definierbare Zeichenfolge festgelegt. Diese kann auch Variable, wie Datum, Uhrzeit, User-Id und Name des angeloggten Host enthalten (Standard: "/").

param Zeichenfolge für den BS2000-Prompt in Connections. *param* darf 60 Zeichen nicht überschreiten. Es findet keine Umsetzung von Kleinbuchstaben in die entsprechenden Großbuchstaben statt.

Außer Konstanten können in *param* auch Variable verwendet werden. Diese werden zum Zeitpunkt der Ausgabe des Prompts durch die dann gültigen aktuellen Werte ersetzt.
Der Name einer Variablen wird abgeschlossen durch ein Sonderzeichen, z.B. Blank oder Komma. Falls nach der Variablen ein Buchstabe oder eine Ziffer als konstantes Zeichen ausgegeben werden soll, so ist der Name der Variablen durch einen Punkt abzuschließen. Der Punkt erscheint nicht im Text.

Folgende Variablen sind möglich:

&CON Nummer der Connection, in der sich der Benutzer gerade befindet.

&USER-ID User-Id, unter der sich der Benutzer in der aktuellen Connection angeloggt hat.

&LOGNAM Jobname des LOGON-Kommandos für die Connection.

&HOST Name des Hostrechners, unter dem der Dialogprozeß läuft.

&DATE Tagesdatum in der Form tt.mm.yy

&TIME aktuelle Uhrzeit in der Form hh:mm

&(*name*) Inhalt der angegebenen Job- oder SDF-P Variablen

Hinweise:

Der Connection-Prompt wirkt in allen Connections zu $DIALOG, sofern nicht mit der SDF-Option GUIDE=NO gearbeitet wird.

Bei geschachtelten Connections (siehe Seite 352) wird stets der Connection-Prompt des zuletzt geladenen CFS wirksam.

Falls das CP-Kommando in einer Startup-Datei enthalten ist (siehe Kapitel 20 "Startup") und Variable verwendet werden, so sollte die folgende Kommandofolge angegeben werden, um Konflikte mit CFS-Prozedurvariablen zu vermeiden:

```
*SET PAR=#
CP ' ... '
*SET PAR=&
```

Falls das CP-Kommando in einer Startup-Datei enthalten ist, so können im Hexadezimalmodus des EDT auch Steuerzeichen des Extended Line-Modus in den String des CP-Kommandos eingesetzt werden. Für die Codes dieser Steuerzeichen siehe Manual "Makroaufrufe an den BS2000-Ablaufteil", Makro VTCSET.

Beispiele:

```
CP 'x&CON y&HOST $&USER-ID &TIME x/'
```

x steht im obigen Beispiel für das Line-Mode Steuerzeichen X'14' (EM4) und y für das Zeichen X'1F' (EM2). Aufgrund des CP-Kommandos könnte in einer Connection 1 z.B. der folgende Prompt ausgegeben werden: `1 VAR2 $TSOS 10:22 /`

```
CP 'Conn. &CON  &USER-ID &TIME /'
```

Falls sich der Benutzer in Connection 1 unter der User-Id TEST angeloggt hat, so könnte z.B. folgende Eingabeaufforderung ausgegeben werden:

```
Conn. 1  TEST 13:22 /
```

Wiederherstellen des Originalbildschirms bei Wechsel der Connection

<u>CR</u> | NCR   Connection Refresh / No Connection Refresh.

CR   Nach Verzweigen in eine Connection durch CC*n* (Continue in Connection) wird der Bildschirm wieder in den Zustand gebracht, wie er vor Verlassen der Connection bestand. Der Connection Refresh-Modus erleichtert besonders den Verkehr mit UTM-Anwendungen, die mit Teilmasken arbeiten und keinen eigenen Transaktionscode zum Wiederherstellen einer zerstörten Bildschirmmaske bieten (ähnlich der K3-Taste des EDT). Der Connection-Refresh wirkt nicht bei Connections, für die der Asynchron-Modus eingeschaltet ist (AC*n*).

NCR   Bei Rückkehr in eine Connection mit CC*n* wird bei Masken lediglich die letzte Teilmaske wiederholt. Im Line-Modus wird die letzte Ein-/Ausgabe wiederholt.
Standard: CR

Hinweis:

Um beim automatischen Eröffnen mehrerer Connections mit Hilfe einer Startup-Datei Probleme zu vermeiden, kann es notwendig sein, den CR-Modus vor Eröffnen der ersten Connection auszuschalten (NCR) und nach Eröffnen der letzten Connection wieder einzuschalten.

Namen der CFS-Kommandos für Connections ändern

CRD *bez*=*bez-neu* Connection-Command Redefine.
Die innerhalb von Connections angebbaren Kommandos und Symbole mit einer Sonderbedeutung für CFS wie z.B. LOG/RES/HC, Zeichen für Kommandogedächtnis, Connection-Fluchtsymbol können geändert werden. Die neuen Bezeichnungen gelten global für alle Connections.

Eine Notwendigkeit, die Bezeichnungen der für den Connection-Handler bestimmten Kommandos umzudefinieren, kann sich z.B. ergeben, falls in einer UTM-Anwendung ein Transaktionscode mit einem Kommando für CFS übereinstimmt (z.B. RES).

bez   bisheriger Kommando-Name bzw. Symbol für Kommandogedächtnis, Connection-Fluchtsymbol.

Folgende Kommandos und Symbole können unter *bez* angegeben und durch *bez-neu* redefiniert werden: LOG / NLOG / RES / HC / NHC / STEP / ^ (Connection-Fluchtsymbol) / - (Zeichen für Kommandogedächtnis).

bez-neu   neuer Kommando-Name bzw. neues Symbol für Kommandogedächtnis, Connection-Fluchtsymbol. Bei der Redefinition eines Kommando-Namens ist darauf zu achten, daß der neue Name die gleiche Anzahl Zeichen enthält wie der bisherige Kommando-Name (siehe CRD-Maske auf der folgenden Seite).

Beispiele:

`CRD -=§`	Zeichen für Kommandogedächtnis ändern
`CRD LOG=LLL`	Kommando für Beginn des Loggings ändern
`CRD NLOG=NLLL`	Kommando zum Beenden des Loggings ändern
`CRD RES=RRR`	Kommando zum Wiederholen eines Dialogs ändern
`CRD HC=HH`	Kommando für Beginn des Hardcopies ändern
`CRD NHC=NHH`	Kommando zum Beenden des Hardcopies ändern

CRD Kommando ohne Parameter: Es wird eine Maske ausgegeben, in der die Namen der innerhalb von Connections angebbaren Kommandos und Symbole wie z.B. LOG/ RES/HC, Zeichen für Kommandogedächtnis, Connection-Fluchtsymbol geändert werden können. Die neuen Bezeichnungen gelten global für alle Connections.

Format der CRD-Maske:

```
                   C F S   Connection-Command  Redefine  (CRD)


 You may alter names of commands and command symbols with a special meaning in
 Connections of CFS     (OC0, ... ,OC9)

  Escape-Symbol                 :  ^
  Symbol for Command-Memory     :  -
  Start Logging                 :  LOG           End Logging          :  NLOG
  Start Hardcopy                :  HC            End Hardcopy         :  NHC
  Start Restore of Dialog       :  RES

  during LOG:  STEP on/off      :  STEP

  Suppress Output in Conn.      :  O OFF         No Suppress Output   :  O ON
```

Hinweise:

Der Systemverwalter kann über eine wahlweise Option CFS so generieren (siehe CFSMAIN), daß das Fluchtsymbol ^ vor allen in der obigen Maske aufgeführten Kommandos angegeben werden muß. Das Kommandogedächtnis wird dann nicht durch "-", sondern durch "^-" aktiviert. Die LOG-, bzw. RES-Funktion wird in diesem Fall durch ^LOG, bzw. ^RES eingeschaltet.

Falls die genannte CFS-Option durch den Systemverwalter aktiviert wurde, so wird dies in der CRD-Maske durch das Fluchtsymbol ^ vor den entsprechenden Kommandos dargestellt. Z.B.:

```
Command-Memory     :  -       -->  Command-Memory     :  ^-
Start Logging      :  LOG     -->  Start Logging      :  ^LOG
```

K2-Taste direkt an Connection weiterleiten

DKC*n*

Direct K2-mode for Connection *n*.
Die Betätigung der K2-Taste wird unmittelbar an die Connection weitergeleitet. In diesem Modus verliert die K2-Taste die Funktion, den Übergang in eine andere Connection bzw. in die Grundebene von CFS zu steuern. Siehe hierzu Seite 349. Die normalerweise in CFS-Connections bei Betätigung der K2-Taste ausgegebene Meldung "Return to CFS/Continue in Conn. n/Break ? (R/n/n cmd/B/^cmd/?)" wird in diesem Fall unterdrückt. Es wird grundsätzlich so verfahren, als ob obige Meldung mit der Eingabe B (Break) beantwortet worden wäre.

Hinweise:

Zur Eingabe von Kommandos für bestimmte Programme, wie z.B. TELNET oder Software-Pad, muß der DKC-Modus zuvor zwingend eingeschaltet werden.

Bei eingeschaltetem DKC-Modus kann die betreffende Connection nur bei Eingabeaufforderung durch Eingabe des Fluchtsymbols (siehe Seite 351) in Richtung einer anderen Connection bzw. in Richtung CFS-Grundebene verlassen werden.

Der DKC-Modus wirkt nicht bei Connections zu $CONSOLE. In diesem Fall wird die K2-Taste stets als Anruftaste zur Einleitung einer Konsoleingabe verwendet.

NDKC*n* (Standard)

No Direct K2-mode for Connection *n*.
Bei Betätigung der K2-Taste in einer unter CFS eröffneten Connection wird der Benutzer gefragt, ob er in eine andere Connection bzw. in die Grundebene von CFS verzweigen möchte oder ob die K2-Taste an den Kommunikationspartner weitergeleitet werden soll.
Standard: NDKC*n*.

CFS Connection-Exit ausschalten

NEC*n*

No Exit for Connection *n*.
Die Eingaben für die Connection *n* laufen nicht über den Connection-Exit von CFS. Dies hat zur Folge, daß in der Connection alle CFS-spezifischen Kommandos, wie LOG/NLOG/RES/HC/NHC/O [ON|OFF]/STEP [ON|OFF] nicht ausgewertet werden. Bereits eingeschaltete LOG-/RES-Funktionen werden nicht weitergeführt. Die Auswertung des Fluchtsymbols wird durch das Kommando NEC nicht beeinflußt.

Der NEC-Modus kann bei UTM-Anwendungen nützlich sein, in denen die Eingaben LOG/NLOG/RES/HC als Transaktionscodes benützt werden und nicht vom Connection-Handler als interne Kommandos zu interpretieren sind.

In diesem Zusammenhang wird auch auf das Kommando CRD (Connection-Command Redefine) hingewiesen. Mit diesem Kommando kann die Bezeichnung für jedes CFS-spezifische Connection-Kommando wie LOG/NLOG/RES/HC/NHC/... einzeln umdefiniert werden.

EC*n* (Standard)

Exit for Connection *n*.
Die Eingaben für die Connection *n* laufen über den Connection-Exit. LOG-/ RES- und alle weiteren vom Connection-Handler des CFS angebotenen Funktionen werden in diesem Modus ausgeführt.
Standard: EC*n*, d.h. der Exit für Connection *n* ist eingeschaltet.

Connection-Status in Zeile 25 einblenden

SCL | NSCL

Show Connection Line.
Befindet sich der Benutzer in einer Connection, so wird in der Systemzeile des Bildschirms eine Übersicht der wichtigsten Parameter der Connection, wie z.B. die Nummer der Connection, der Name der Anwendung, des virtuellen Terminals, des Hostrechners, sowie bei $DIALOG zusätzlich die Benutzerkennung des LOGON-Kommandos und die TSN angezeigt. Beim Wechsel der Connections wird der Inhalt der Systemzeile angepaßt.
Standard: NSCL

Hinweis:

Durch die Connection-Zeile erhält der Benutzer jederzeit den Überblick über die gerade aktive Connection. Dies ist von besonderem Vorteil für Endanwender, die mit dem Menue-System arbeiten.

Trace Connection (Diagnose)

TC*n* [*datei*]

Trace Connection *n*.
TC wird in der Grundebene von CFS im Kommandofeld eingegeben und bewirkt daß alle für Diagnosezwecke relevanten DCAM-Nachrichten der Connection *n* in einer Datei protokolliert werden. Das Kommando kann bereits vor Eröffnen der Connection abgesetzt werden.

datei

Name der Protokolldatei. Falls kein Name angegeben ist, wird der Standardname CFS.CONNECTION.TRACE verwendet.

Hinweis:

Das Kommando TC kann mehrfach für verschiedene Connections eingegeben werden. Der Dateiname darf in diesem Fall jedoch nur bei dem ersten TC-Kommando mit angegeben werden.

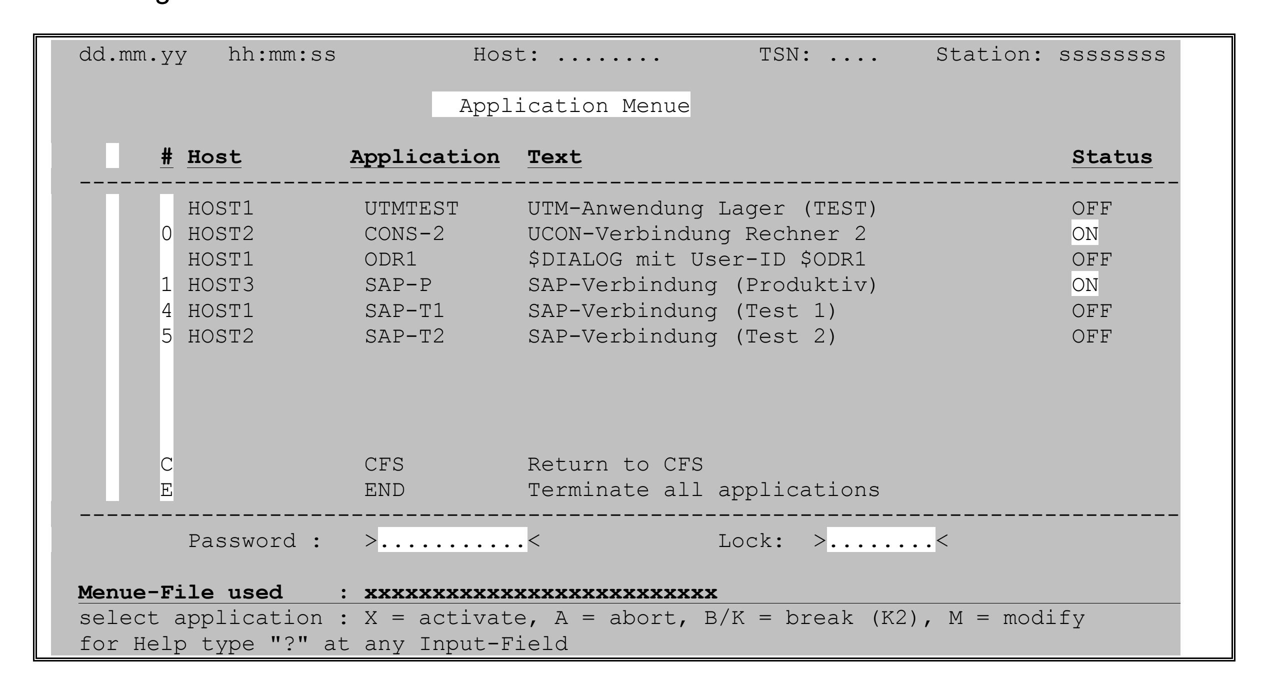
NTC*n*

No Trace for Connection *n*.
Der Trace für die Connection *n* wird beendet.
Falls der Trace für mehrere Connections eingeschaltet war, muß für jede Connection ein NTC-Kommando abgesetzt werden, damit die Trace-Datei geschlossen und für den Benutzer zugreifbar wird.

10. Menue-System für Connections

Im vorhergehenden Kapitel (9. Connections) wurden die Kommandos zum Administrieren von Connections dargestellt: OC zum Eröffnen von Connections, CC zur Wiederaufnahme des Dialogs mit einer Connection, DC zum Beenden einer Connection und CD zur Anzeige einer Übersicht aller eröffneten Connections.

Über das Kommando **MENUE** bietet CFS einen maskengeführten und auf einer höheren Ebene angesiedelten Zugang zu den Connections. Für den Benutzer ist es dabei nicht mehr notwendig, die einzelnen Kommandos zum Administrieren der Connections zu kennen. Durch einfaches Markieren einer in der Menue-Maske vordefinierten Connection wird diese von CFS selbständig eröffnet bzw. der Dialog mit dieser Connection fortgesetzt, d.h. CFS setzt selbst die erforderlichen Kommandos zum Eröffnen bzw. Fortsetzen des Dialogs ab.



```
dd.mm.yy   hh:mm:ss         Host: ........      TSN: ....    Station: ssssssss

                              Application Menue

     # Host        Application  Text                                    Status
--------------------------------------------------------------------------------
       HOST1        UTMTEST     UTM-Anwendung Lager (TEST)              OFF
     0 HOST2        CONS-2      UCON-Verbindung Rechner 2               ON
       HOST1        ODR1        $DIALOG mit User-ID $ODR1               OFF
     1 HOST3        SAP-P       SAP-Verbindung (Produktiv)              ON
     4 HOST1        SAP-T1      SAP-Verbindung (Test 1)                 OFF
     5 HOST2        SAP-T2      SAP-Verbindung (Test 2)                 OFF




     C              CFS         Return to CFS
     E              END         Terminate all applications
--------------------------------------------------------------------------------
        Password :   >...........<            Lock:  >........<

Menue-File used    : xxxxxxxxxxxxxxxxxxxxxxxxxx
select application : X = activate, A = abort, B/K = break (K2), M = modify
for Help type "?" at any Input-Field
```

Das Menue-System für Connections bietet dem Benutzer folgende Dienste:

1) Benutzerspezifische Zusammenstellung von vordefinierten Connections.
   In einer sog. Menue-Datei können die Parameter von Connections festgelegt werden, mit denen der Benutzer nachfolgend möglicherweise arbeiten möchte. Zu den Parametern der Connection zählen u.a. Name der Anwendung, Host-Rechner, Name des virtuellen Terminals, LOGON-/KDCSIGN-Kommando, Startup-Eingaben.
   In der Standardmaske des Menue-Systems können bis zu zehn Connections vordefiniert werden. Eine besondere Hintergrundmaske bietet Platz für weitere zehn Connections, die bei Bedarf in die Hauptmaske übernommen werden können.

2) Durch Ankreuzen einer im Menue vordefinierten Connection wird diese, gemäß den in der Menue-Datei gespeicherten Parametern, selbständig eröffnet. Gegebenenfalls wird nach dem Eröffnen eine Startsequenz von Kommandos automatisch ausgeführt. Ist die vom Benutzer angekreuzte Connection bereits eröffnet, so wird der Dialog mit dieser einfach fortgesetzt.

3) Automatisches Verwalten der Connection-Nummern. Beim Eröffnen eines neuen Verfahrens wird automatisch die Nummer der nächsten freien Connection ermittelt.

4) Übersicht über die bereits eröffneten bzw. noch zu eröffnenden Connections zusammen mit ihrem Status (ON/OFF) in Form einer übersichtlichen Maske. Durch Drücken der K2-Taste innerhalb einer Connection wird automatisch in die Menue-Maske verzweigt. Bei eingeschaltetem DKC-Modus (Direct K2 for Connection, siehe Seite 376) kann nur durch Eingabe des Fluchtsymbols in die Menue-Maske verzweigt werden.

5) Für jedes in der Menue-Maske aufgeführte Verfahren kann eine P-Tastenbelegung (PLUS) festgelegt werden. Beim Wechsel zwischen den Verfahren (Connections) werden die P-Tasten entsprechend umgeladen.

6) Die in der Menue-Maske vordefinierten Connections können vom Benutzer mit einem Menue-Editor im Online-Betrieb jederzeit verändert, ergänzt oder in der Reihenfolge umgestellt werden. Es besteht die Möglichkeit, nur einzelnen Benutzern das Recht zur nachträglichen Veränderung der Menue-Datei einzuräumen.

Für den im BS2000 und CFS unerfahrenen Benutzer (z.B. Sachbearbeiter) bietet das Menue-System den geeigneten Zugang zu den Multi-Connection Services von CFS, die auch für diesen Anwenderkreis von großem Nutzen sind. Nach entsprechender organisatorischer Vorbereitung muß der Benutzer nur ein für ihn vorgesehenes LOGON-Kommando eingeben. Er erhält danach automatisch eine auf seine Bedürfnisse zugeschnittene Menue-Maske mit der Wahl zwischen verschiedenen, für den Benutzer relevanten und parallel abwickelbaren Verfahren, die über Connections von CFS realisiert sind.

Zu den organisatorischen Vorarbeiten, die normalerweise der Systemverwalter durchzuführen hat, zählen:

a) Erstellen einer auf den Anwender zugeschnittenen Menue-Datei, evtl. mit Startup-Sequenzen für die zu eröffnenden Connections.

b) Erstellen einer LOGON-DO Prozedur SYS.SDF.LOGON.USERPROC unter der Kennung des Benutzers für den automatischen Start von CFS nach dem LOGON-Kommando. Voraussetzung hierfür ist BS2000 mit SDF ab Version 1.4. Die LOGON-DO Prozedur muß die BS2000-Kommandos bis zum Programmaufruf von CFS, sowie die CFS-interne Kommandofolge bis zum Aufruf des Kommandos MENUE enthalten. Hierzu wird auf Seite 485 und folgende (Prozedursprache) und 537 (CFS-Startup) verwiesen.

Aufruf der Menue-Maske

MENUE [*datei*] [,N]

Die Menue-Maske wird aufgerufen durch das Kommando MENUE [*datei*] [,N].
MENUE kann abgekürzt werden als MEN oder MENU.
Als Basisinformation für den Aufbau der Menue-Maske wird nach einer Menue-Datei gesucht, sofern diese nicht im Kommando selbst angegeben wurde. In der Menue-Datei sind alle Angaben zu den Connections (Verfahren) gespeichert, mit denen der Benutzer im folgenden arbeiten kann.

Durch den Parameter N wird eine leere Menue-Datei angelegt, falls eine Datei des angegebenen Namens nicht existiert.

Durch Ankreuzen eines im Menue aufgeführten Verfahrens gelangt der Benutzer direkt in die dafür vorgesehene Connection. Die Rückkehr in die Menue-Maske erfolgt durch Betätigen der K2-Taste. Bei eingeschaltetem DKC-Modus (Direct K2 for Connection) wird die K2-Taste direkt an die Connection weitergeleitet. In diesem Fall kann der Benutzer nur durch Eingabe des Fluchtsymbols ^ in die Menue-Maske zurückkehren.

Das direkte Wechseln zwischen bereits eröffneten Connections mit dem Fluchtsymbol (^*n*) ist auch unter Steuerung des Menue-Systems möglich.

Das Menue-System kann in Richtung CFS verlassen werden, sofern das Verfahren CFS für den Benutzer durch einen entsprechenden Eintrag in der Menue-Datei zugelassen ist. Durch Eintragen des Kurzkommandos 'C' in der ersten Eingabespalte des Menues oder durch Betätigen der K1-Taste gelangt der Benutzer zurück in die CFS-Maske. Auch hier stehen dem Benutzer alle durch das Menue-System eröffneten Connections zur Verfügung.

Die oben erwähnte CFS-Berechtigung gestattet es dem Menue-Benutzer auch, den Menue-Editor aufzurufen (Eintrag von 'M' im Eingabefeld eines Verfahrens oder im globalen Eingabefeld des Menues). Im Menue-Editor können die Angaben zu den einzelnen Verfahren geändert werden. Es können auch neue Menue-Einträge angelegt bzw. vorhandene gelöscht werden. Beim Verlassen des Menue-Systems wird der Benutzer gefragt, ob die vorgenommenen Änderungen in die Menue-Datei übernommen werden sollen.

Hierarchie der verwendeten Menue-Dateien

Die vom Menue-System zu verwendende Menue-Datei wird nach der folgenden Hierarchie bestimmt:

1) Wurde im Kommando MENUE ein Dateiname angegeben ?

2) Existiert ein File-Kommando mit LINK=MENUE (/FILE ...,LINK=MENUE) ?

3) Existiert kein TFT-Eintrag mit dem Link-Namen MENUE, so wird der im LOGON-Kommando angegebene Job-Name gesucht (/.*name* LOGON ...) und an das Prefix 'CFS.MENUE.' angehängt. Existiert eine Datei CFS.MENUE.*name*, so wird diese als Menue-Datei verwendet.

4) Wurde beim LOGON-Kommando kein Jobname angegeben oder existiert die Datei CFS.MENUE.*name* nicht, so wird unter der eigenen Benutzerkennung eine Datei CFS.MENUE gesucht.

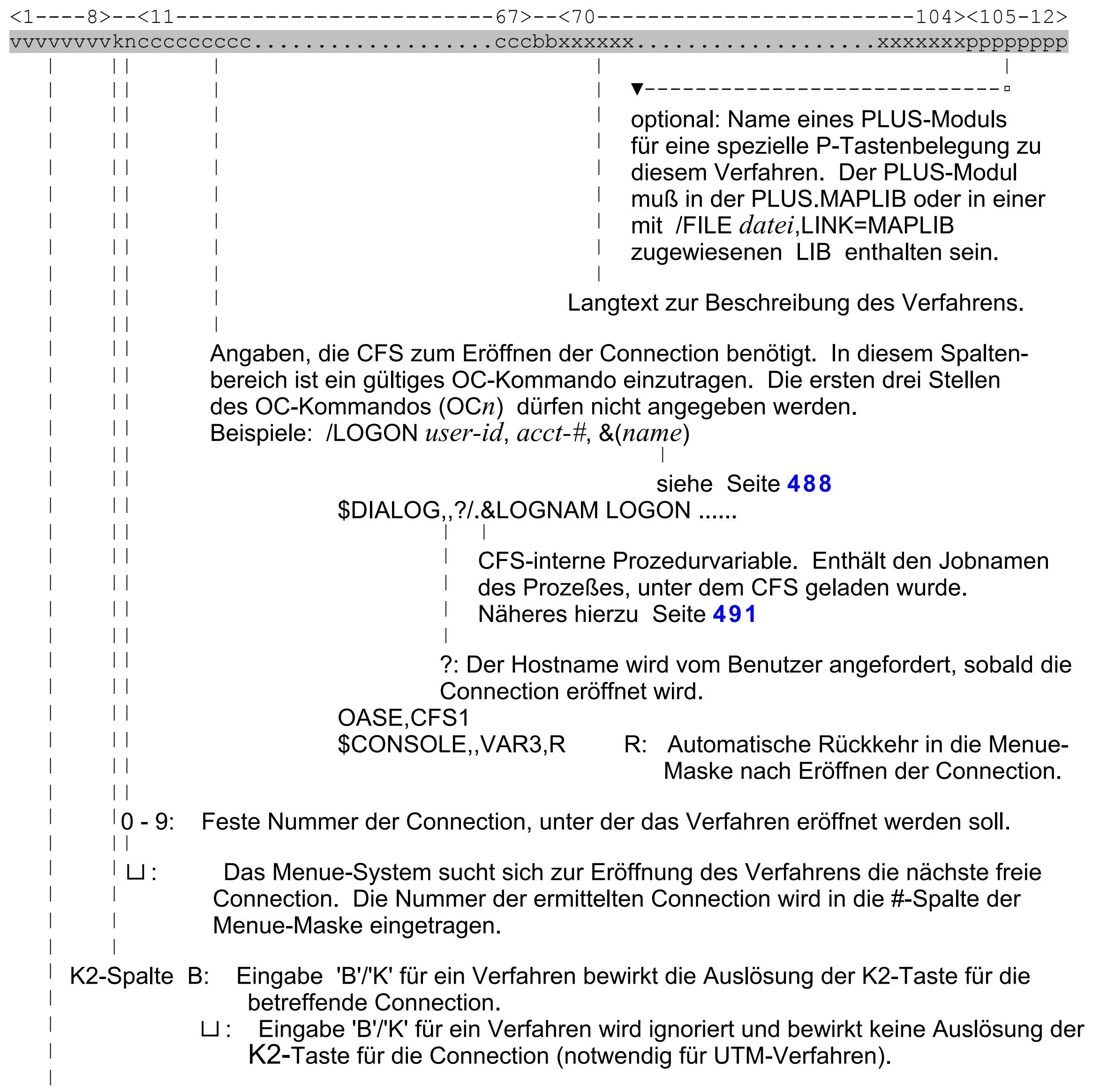
5) Existiert keine der oben genannten Dateien, so sucht CFS nach einer Datei $*user-id*.CFS.MENUE.*name* . $*user-id* ist die Benutzerkennung, unter der das Programm CFS gespeichert ist. *name* ist der beim LOGON-Kommando angegebene Jobname (siehe 3)).

6) Als letztes wird nach einer Datei $*user-id*.CFS.MENUE gesucht.

Format der Menue-Datei

Die Menue-Datei ist eine sequentielle Datei mit den für den Benutzer vordefinierten und über Connections von CFS realisierten Verfahren.

Die ein Verfahren beschreibenden Datensätze der Menue-Datei besitzen folgendes Format:



Es können bis zu 20 Datensätze der beschriebenen Art in der Menue-Datei angegeben werden. Die ersten 10 werden im Hauptmenue angezeigt. Über die Eingabe 'M' in der Auswahlspalte gelangt der Benutzer in den Menue-Editor. Hier werden alle 20 Verfahren im Überblick dargestellt. Die Reihenfolge der Verfahren 1 bis 20 kann neu festgelegt werden. Z.B. kann ein im Moment nicht benötigtes Verfahren mit der bisherigen Nummer 9 durch das Verfahren 17 ersetzt werden.

Hinweise:

Die Verzweigung aus der Menue-Maske in das Programm CFS ist für den Benutzer nur möglich, falls ein zusätzlicher Datensatz in die Menue-Datei aufgenommen wird, der in der Verfahrensspalte die Zeichenfolge CFS und außer in Spalte 105 - 112 überall Leerzeichen enthält. In Spalte 105 - 112 kann eine optionale P-Tastenbelegung für das Programm CFS festgelegt werden. Existiert in der Menue-Datei kein derartiger Satz, so erscheint der Menuepunkt 'C' (CFS) nicht in der Menue-Maske und der Benutzer kann nicht in das Programm CFS verzweigen. In diesem Fall kann das Menue-System einschließlich CFS nur durch Ankreuzen des letzten Menuepunktes 'E' (ENDE) verlassen werden.

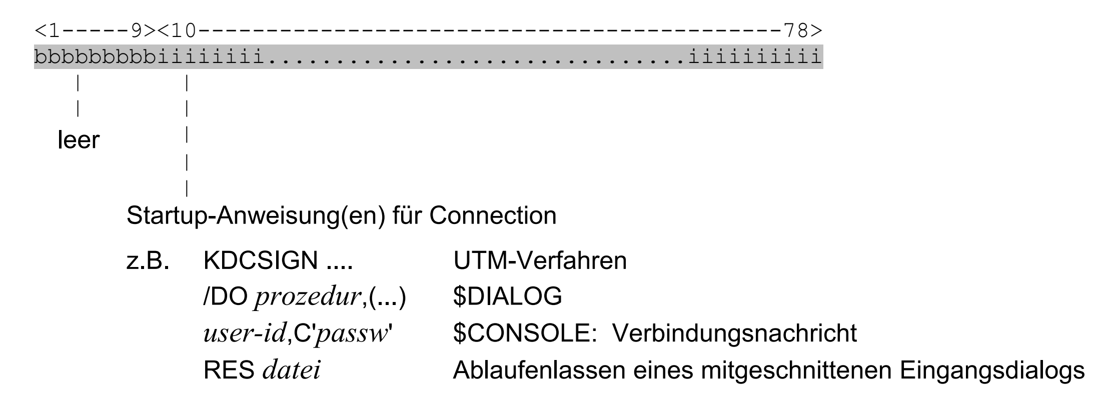
Die Möglichkeit, in das Programm CFS zu verzweigen, berechtigt den Benutzer auch, den Menue-Editor zur Veränderung, Neuaufnahme und zur Löschung von Verfahrenseinträgen (Menuepunkten) zu verwenden.

Mit Ausnahme des Verfahrens 'CFS' ist das Einfügen der Datensätze für alle anderen Verfahren in die Menue-Datei sowohl mit EDT, als auch mit dem speziellen Menue-Editor von CFS (Eingabe 'M' Modify, siehe unten) möglich. Das Erstellen oder Verändern der Einträge der Menue-Datei mit dem Menue-Editor setzt jedoch die Berechtigung zur Verzweigung in das Programm CFS voraus. Die Menue-Datei muß also bereits existieren und zumindest den Verfahrenssatz 'CFS' enthalten.

Bei Verbindungen zu $DIALOG kann im OC-Kommando und folglich auch im Beschreibungssatz für das Verfahren ein LOGON-Kommando mitgegeben werden. Zwei Beispiele dafür wurden oben angeführt. Falls in dem im Beschreibungssatz angegebenen LOGON-Kommando kein LOGON-Passwort enthalten ist, obwohl für den Zugang zu der angegebenen Benutzerkennung ein Passwort erforderlich ist, so muß dies beim Eröffnen des Verfahrens im Feld 'Passwort' der Menue-Maske nachgereicht werden. Beschreibung der Menue-Maske: siehe unten.

Im Anschluß an den ein Verfahren beschreibenden Datensatz können in der Menue-Datei ein oder mehrere Datensätze vorgegeben werden, die als automatische Startup-Eingaben nach dem erfolgreichen Eröffnen der Connection an den Partner gesendet werden.

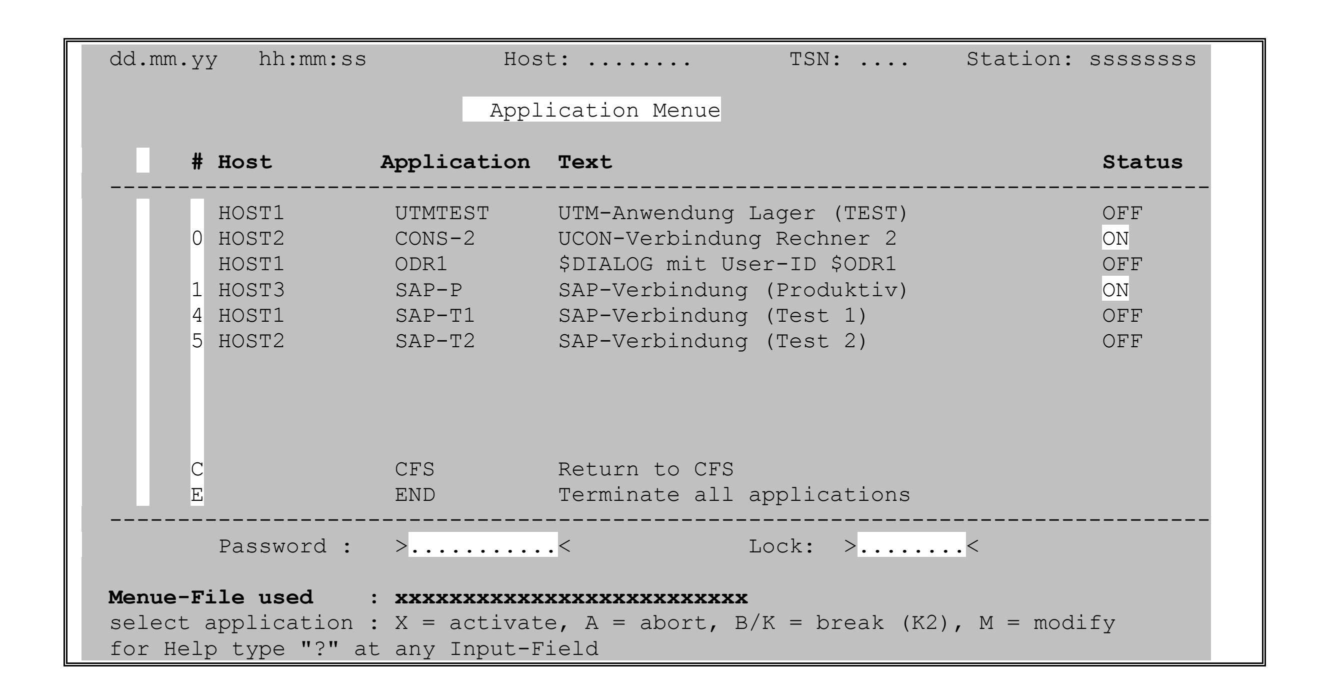
Format der Startup-Sätze:



Es können beliebig viele Startup-Sätze zu einem Verfahren angegeben werden.

Als Startup-Sätze können auch beliebige Anweisungen aus der CFS-Prozedursprache (*-Anweisungen) angegeben werden. Auch CFS-interne Prozedurvariable können verwendet werden. Für eine ausführliche Beschreibung siehe Seite 485 und folgende (CFS-Prozedursprache).

Format der Menue-Maske



```
dd.mm.yy   hh:mm:ss         Host: ........       TSN: ....    Station: ssssssss

                              Application Menue

       # Host        Application  Text                                Status
---------------------------------------------------------------------------------
         HOST1        UTMTEST     UTM-Anwendung Lager (TEST)          OFF
       0 HOST2        CONS-2      UCON-Verbindung Rechner 2           ON
         HOST1        ODR1        $DIALOG mit User-ID $ODR1           OFF
       1 HOST3        SAP-P       SAP-Verbindung (Produktiv)          ON
       4 HOST1        SAP-T1      SAP-Verbindung (Test 1)             OFF
       5 HOST2        SAP-T2      SAP-Verbindung (Test 2)             OFF




       C              CFS         Return to CFS
       E              END         Terminate all applications
---------------------------------------------------------------------------------
        Password :   >..........<             Lock:  >........<

Menue-File used     : xxxxxxxxxxxxxxxxxxxxxxxxxx
select application : X = activate, A = abort, B/K = break (K2), M = modify
for Help type "?" at any Input-Field
```

Beschreibung der Felder der Maske:

Bei 'Menue : *xxxxxxxxxxxxx*' wird der Name der verwendeten Menue-Datei angezeigt. Im Feld 'Station : *ssssssss*' wird die Bezeichnung der Datensichtstation ausgegeben, von der das Programm CFS aufgerufen wurde. Die Angaben *xxxxxxxx* und *ssssssss* können z.B. für den Systemverwalter bei Anfragen von Sachbearbeitern von Interesse sein.

Für jede in der Menue-Datei vordefinierte Connection wird im Menue eine Zeile ausgegeben. Diese besteht aus einem einstelligen **Eingabefeld**, einer vorgegebenen oder bei Eröffnen der Connection einzutragenden Connection-Nummer (#-Spalte), dem Namen des Host-Rechners, in dem die Connection eröffnet wird, einer Kurzbezeichnung für das Verfahren bzw. für die Connection, sowie einer verbalen Beschreibung des Verfahrens. Die Status-Spalte wird von CFS mit ON/OFF besetzt, je nachdem, ob die betreffende Connection eröffnet ist oder nicht.

Im **globalen Eingabefeld** in der Überschriftszeile kann die einem Verfahren zugeordnete Nummer (Connection-Nummer), siehe #-Spalte angegeben werden. Hiermit wird entweder in die bereits eröffnete Connection verzweigt oder es wird eine neue Connection mit dem gewünschten Verfahren unter der angegebenen Nummer eröffnet. Außer einer Connection-Nummer kann im globalen Eingabefeld auch einer der Codes C (Rückkehr nach CFS), E (Beendigung aller Verfahren einschließlich des Programms CFS), M (Verzweigen in das Modify Menue) oder S (Save) angegeben werden. Eine genaue Beschreibung der einzelnen Codes folgt weiter unten.

Wird der Code X in der Zeile "CFS Programm CFS" eingegeben, so bewirkt dies das Verlassen des Menue-Systems und die Rückkehr in die Kommandozeile von CFS. Mit dem Kommando CC (Continue in Connection) kann in jede bereits eröffnete Connection verzweigt werden.
Die Möglichkeit, in das Programm CFS zu wechseln (CFS-Berechtigung), muß für den Benutzer über einen Eintrag in der Menue-Datei (siehe Seite 383) zugelassen werden. Mit der CFS-Berechtigung hat der Benutzer außerdem die Möglichkeit, mit dem Menue-Editor die Angaben zu einzelnen Menue-Punkten zu ändern, sowie Menue-Punkte zu kopieren und zu löschen. Diese Änderungen, die zuerst

nur temporär ausgeführt werden und mit Beendigung von CFS verloren gehen, können auf Wunsch auch in die Menue-Datei zurückgeschrieben werden.

Das Markieren der Zeile "ENDE   Beenden aller Verfahren" mit X bewirkt das Verlassen des Menue-Systems und die Beendigung des Programms CFS. Die Funktion ENDE schließt alle eröffneten Verfahren (Connections) mit DC (Disable Connection) und beendet danach das Programm CFS.

Wurde CFS mit einer Startup-Kommandofolge aufgerufen, die die Anweisung '*ON %END CONTINUE' enthält, so werden die eröffneten Connections nicht durch DC beendet. In diesem Fall wird die unterbrochene Startup-Kommandofolge fortgesetzt. Die folgenden, automatisch ausgeführten Kommandos können u.a. dazu benutzt werden, auf kontrollierte Weise noch nicht geschlossene Connections zu beenden (Shutdown-Kommandofolge). Ein Beispiel für eine Startup-Datei mit Shutdown-Kommandofolge finden Sie auf Seite 537 und folgende (Startup-Datei).

Das dunkelgesteuerte Feld 'Passwort' in der letzten Zeile der Menue-Maske dient zur Eingabe des Paßwortes für ein LOGON-Kommando. Das Feld muß nicht ausgefüllt werden, falls das Paßwort bereits im LOGON-Kommando in der Menue-Datei eingetragen ist.

Das dunkelgesteuerte Feld 'Lock' in der letzten Zeile der Menue-Maske dient zum Sperren des Bildschirms bei Abwesenheit des Benutzers. Wird hier ein Paßwort eingetragen, so wird der Bildschirm solange gesperrt, bis der Benutzer die Lock-Funktion durch Eingabe des zuvor vereinbarten Passworts wieder aufhebt.

Globales Eingabefeld in der Überschriftszeile:

n Im globalen Eingabefeld kann die in der #-Spalte angezeigte Nummer einer Connection angegeben werden. Die Wirkung entspricht dem Markieren der entsprechenden Menue-Zeile mit X, d.h. die Connection wird eröffnet (falls Status = OFF) und es wird in die Connection verzweigt.

C Verzweigen in das Programm CFS. Die K1-Taste bewirkt ebenso wie die Eingabe 'C' das Verlassen des Menue-Systems und die Verzweigung in die Kommandozeile von CFS. Über das Kommando MENUE kann die Menue-Maske jederzeit wieder aufgerufen werden.
Durch einen Eintrag in der Menue-Datei kann dem Anwender das "Verfahren" CFS, d.h. die Benutzung der "File Services" von CFS erlaubt oder vorenthalten werden.

E Beenden des Menuesystems und Beenden des Programms CFS. Die Funktion ENDE schließt alle eröffneten Verfahren (Connections) mit DC (Disable Connection) und beendet danach das Programm CFS.
Falls beim Aufruf von CFS eine Startup-Datei mit Shutdown-Kommandofolge verwendet wurde (siehe Seite 537 und folgende), so wird durch die Eingabe 'E' die in der Startup-Datei definierte Shutdown-Kommandofolge ausgeführt. Die eröffneten Connections werden in diesem Fall vom Menue-System nicht durch DC beendet.

M Modify. Es wird in das Modify-Menue verzweigt. Das Modify-Menue bietet Platz für bis zu 20 Einträge und erlaubt das Modifizieren, Kopieren und Löschen einzelner Menuepunkte sowie das Ändern der Reihenfolge. Durch eine Änderung der Reihenfolge können Menuepunkte, die in der Hauptmaske nicht dargestellt sind, ausgetauscht werden mit Menuepunkten der Nummern 0 bis 9, die im Moment nicht benötigt werden.

S Save. Nach der Ausführung von Modifikationen im Menue kann mit 'S' das Zurückschreiben des aktuellen Standes in die Menue-Datei forciert werden. Normalerweise wird der Benutzer erst beim Verlassen des Menue-Systems mit 'C' (CFS) bzw. 'E' (ENDE) gefragt, ob der aktuelle Stand des Menues in die Menue-Datei gespeichert werden soll.

Falls der Benutzer nicht CFS-berechtigt ist (fehlender Eintrag 'CFS' in der Menue-Datei), so werden die Eingaben 'M', 'C' und 'S' in der Markierungsspalte der Überschriftszeile zurückgewiesen.

Im **Eingabefeld der einzelnen Menuezeilen** sind folgende Eintragungen möglich:

A Abbruch. Damit wird ein DC (Disable Connection) generiert. Die Funktion 'A' wirkt wie ein abruptes Ausschalten des Bildschirms und sollte nicht oder nur bei Connections zu $CONSOLE angewendet werden.

Soweit irgend möglich, sollte die normale Beendigung der Verfahren (z.B. durch LOGOFF bei $DIALOG oder KDCOFF bei UTM-Anwendungen) gewählt werden.

B Diese Eintragung ist nur bei Verbindungen zu $DIALOG sinnvoll. 'B' simuliert die Betätigung der K2-Taste für die betreffende Connection. Siehe auch Eingabe K weiter unten.

C Copy. Ein bestehender Menuepunkt wird zum Kopieren vorgemerkt. Das Ziel des Kopiervorgangs wird mit 'T' (To) gekennzeichnet. Das Ergebnis des Kopiervorgangs wird in der Menue-Maske sofort dargestellt. Beim Verlassen des Menue-Systems mit 'E' (ENDE) oder 'C' (CFS) wird der Benutzer gefragt, ob die im Speicher vorgenommenen Änderungen in die Menue-Datei zurückgeschrieben werden sollen.

Falls der Benutzer nicht CFS-berechtigt ist (fehlender Eintrag 'CFS' in der Menue-Datei), so werden die Eingaben 'C' und 'T' in der Markierungsspalte zurückgewiesen.

D Delete. Der bezeichnete Menuepunkt wird gelöscht. Das Ergebnis der Löschung wird in der Menue-Maske sofort dargestellt. Beim Verlassen des Menue-Systems mit 'E' (ENDE) oder 'C' (CFS) wird der Benutzer gefragt, ob die im Speicher vorgenommenen Änderungen in die Menue-Datei zurückgeschrieben werden sollen.

Falls der Benutzer nicht CFS-berechtigt ist (fehlender Eintrag 'CFS' in der Menue-Datei), so werden die Eingaben 'C' und 'T' in der Markierungsspalte zurückgewiesen.

K Gleiche Bedeutung wie B (siehe oben). 'K' simuliert die Betätigung der K2-Taste für die betreffende Connection. Im Unterschied zu 'B' werden alle Ausgaben bis zum nächsten Kommandoprompt ignoriert. K ist z.B. zum Unterbrechen langer FSTAT-Ausgaben sinnvoll.

M Modify. Es wird in den Menue-Editor verzweigt. Hier werden in einer Maske alle in der Menue-Datei enthaltenen Angaben zu dem Menuepunkt angezeigt. Der Benutzer kann die gewünschten Modifikationen vornehmen. Die Rückkehr in die Menue-Maske erfolgt durch die K1-Taste. Die Änderungen für den Menuepunkt werden sofort wirksam. Beim Verlassen des Menue-Systems mit 'E' (ENDE) oder 'C' (CFS) wird der Benutzer gefragt, ob die im Speicher vorgenommenen Änderungen in die Menue-Datei zurückgeschrieben werden sollen.

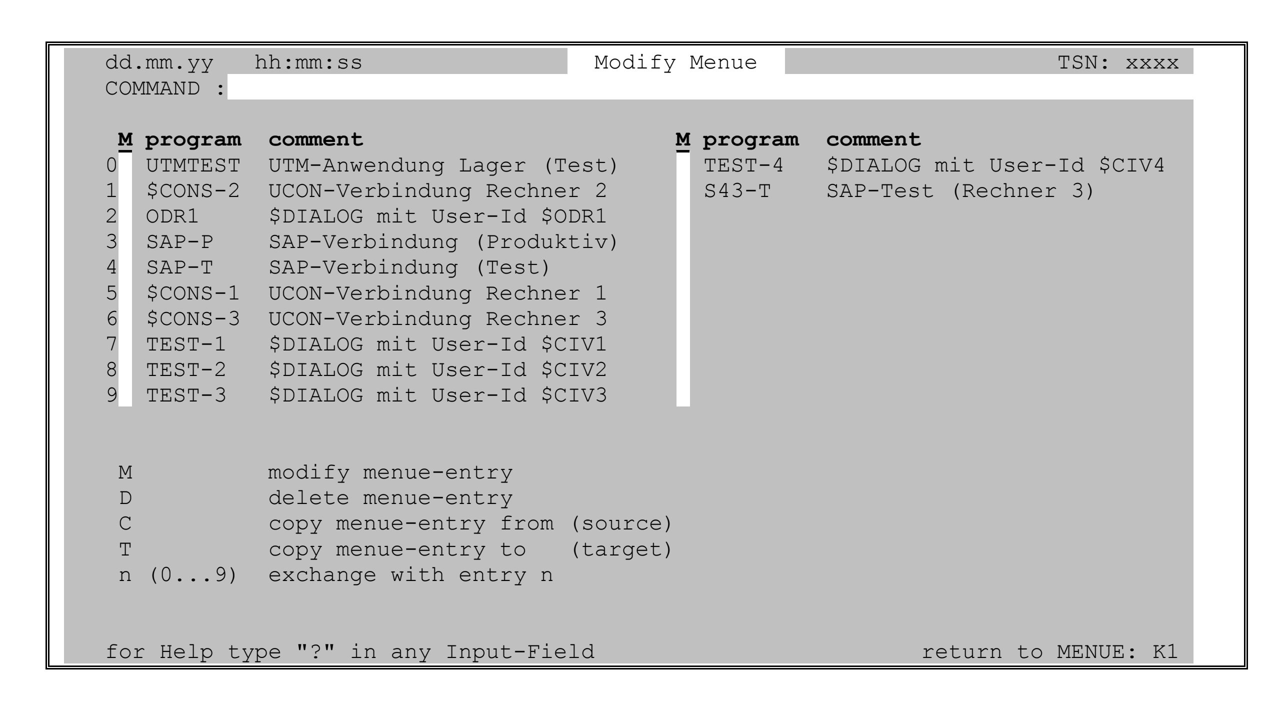
Falls der Benutzer nicht CFS-berechtigt ist (fehlender Eintrag 'CFS' in der Menue-Datei), so wird die Eingabe 'M' in der Markierungsspalte zurückgewiesen.

X Automatisches Eröffnen der Connection, falls Status = OFF.
Verzweigen in den Dialog mit der Connection, falls Status = ON.

Es kann auch mehr als ein Verfahren gleichzeitig mit X markiert werden. In diesem Fall werden die betreffenden Connections eröffnet, der Status auf ON gesetzt und danach wieder in die Menue-Maske verzweigt.

n $0 <= n <= 9$: Austauschen des markierten Menue-Eintrags mit dem Eintrag der Nummer n. Es kann damit die Reihenfolge der Menue-Punkte in der Maske verändert werden.

Die Modify Menue-Maske



```
dd.mm.yy   hh:mm:ss              Modify Menue                       TSN: xxxx
COMMAND :

 M program  comment                          M program  comment
0  UTMTEST  UTM-Anwendung Lager (Test)         TEST-4   $DIALOG mit User-Id $CIV4
1  $CONS-2  UCON-Verbindung Rechner 2          S43-T    SAP-Test (Rechner 3)
2  ODR1     $DIALOG mit User-Id $ODR1
3  SAP-P    SAP-Verbindung (Produktiv)
4  SAP-T    SAP-Verbindung (Test)
5  $CONS-1  UCON-Verbindung Rechner 1
6  $CONS-3  UCON-Verbindung Rechner 3
7  TEST-1   $DIALOG mit User-Id $CIV1
8  TEST-2   $DIALOG mit User-Id $CIV2
9  TEST-3   $DIALOG mit User-Id $CIV3


 M          modify menue-entry
 D          delete menue-entry
 C          copy menue-entry from (source)
 T          copy menue-entry to   (target)
 n (0...9)  exchange with entry n


for Help type "?" in any Input-Field                        return to MENUE: K1
```

Eingaben im **Kommandofeld** der Modify Menue-Maske:

/cmd — beliebiges BS2000-Kommando.

END | **K1** — Rückkehr in die Menue-Maske.

HC | NHC — Hardcopy ein-/ausschalten.

MENUE *datei* — Es wird eine neue Menue-Datei eingelesen. Nach der Rückkehr in die Menue-Maske ist diese Datei die aktuelle Menue-Datei.

F3 — Aufzeichnung der aktuellen Maske in die Hardcopy-Datei von CFS.

In der **Markierungsspalte** (M) zu jedem der 20 Einträge kann eingegeben werden:

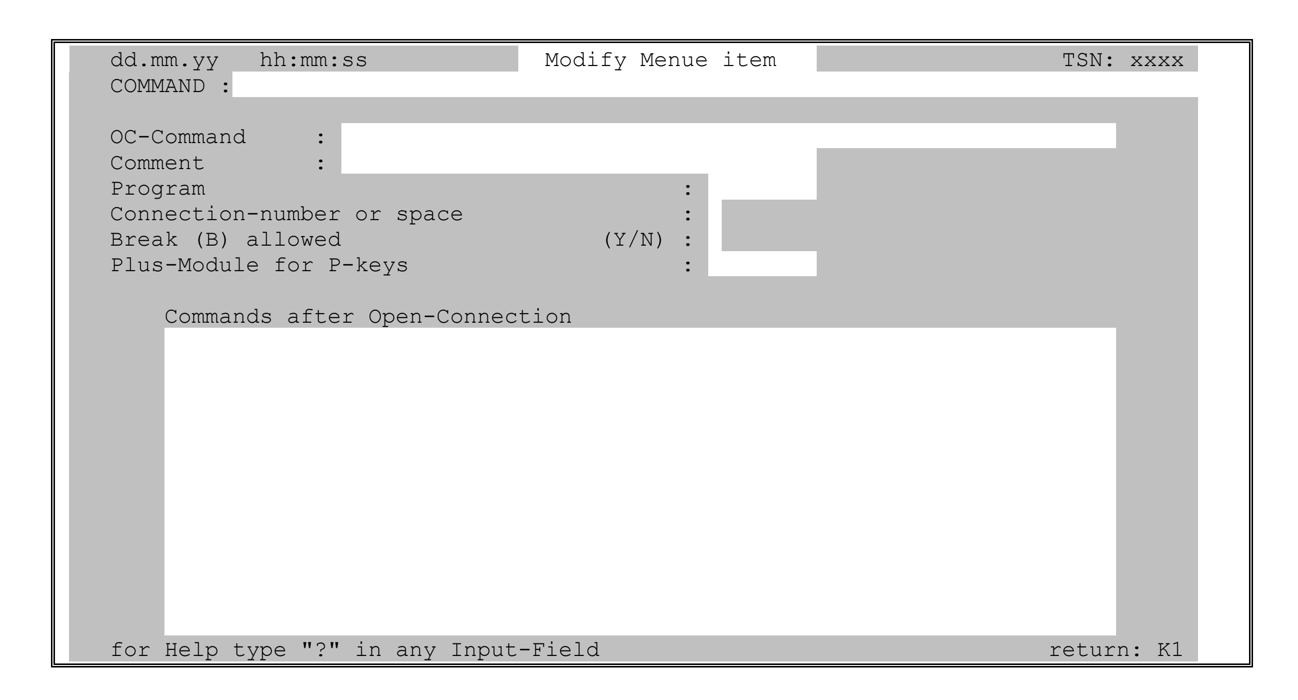
C — Copy. Ein bestehender Menuepunkt wird zum Kopieren vorgemerkt. Das Ziel des Kopiervorgangs wird mit 'T' (To) gekennzeichnet.

D — Delete. Der bezeichnete Menuepunkt wird gelöscht.

M — Modify. Es wird in den Menue-Editor verzweigt. Hier werden in einer Maske alle in der Menue-Datei enthaltenen Angaben zu dem Menuepunkt angezeigt. Der Benutzer kann die gewünschten Modifikationen vornehmen.

n — 0 <= *n* <= 9: Austauschen des markierten Menue-Eintrags mit dem Eintrag der Nummer *n*. Es können damit bislang nicht dargestellte Menue-Punkte 10 bis 19 in eine der Positionen 0 bis 9 übernommen werden. Dies hat zur Folge, daß die betreffenden Menue-Punkte in der Menue-Maske dargestellt werden.

Die Modify Menue-Item Maske



```
dd.mm.yy   hh:mm:ss          Modify Menue item                      TSN: xxxx
COMMAND :

OC-Command      :
Comment         :
Program                                       :
Connection-number or space                    :
Break (B) allowed                       (Y/N) :
Plus-Module for P-keys                        :

    Commands after Open-Connection

for Help type "?" in any Input-Field                                return: K1
```

Eingaben im **Kommandofeld** der Modify Menue Item-Maske:

/cmd	beliebiges BS2000-Kommando.
HC | NHC	Hardcopy ein-/ausschalten.
MENUE *datei*	Es wird eine neue Menue-Datei eingelesen. Diese kann in der gleichen Maske weiter bearbeitet werden.
K1	Rückkehr in die Menue-Maske.
F3	Aufzeichnung der aktuellen Maske in die Hardcopy-Datei von CFS.

Feld "**OC-Command**"

Angaben, die CFS zum Eröffnen der Connection benötigt. In diesem Feld ist ein gültiges OC-Kommando einzutragen. Die ersten drei Stellen des OC-Kommandos (OC*n*) dürfen nicht angegeben werden. Falls in diesem Feld nichts eingetragen ist, so wird eine Connection zu $DIALOG am eigenen Host eröffnet. Das LOGON-Kommando muß vom Benutzer per Hand nach Eröffnen der Connection oder im Maskenfeld "Commands after Open-Connection" angegeben werden.

Beispiele:

`$DIALOG,,VAR2/.&LOGNAM LOGON ......`
&LOGNAM: CFS-interne Prozedurvariable. Enthält den Jobnamen des Prozesses, unter dem CFS geladen wurde.

`/LOGON user-id,acct-#,&(name)`
Das LOGON-Passwort ist in der Jobvariablen jv-name hinterlegt.

`$CONSOLE,,VAR3,R`
R: Automatische Rückkehr in die Menue-Maske nach Eröffnen der Connection.

Feld "**Comment**"

Hier kann kann ein Langtext zur Beschreibung des Verfahrens angegeben werden, der in der Beschreibungsspalte der Menue-Maske erscheint. Eingegebene Kleinbuchstaben werden unverändert übernommen.
Falls in diesem Feld nichts angegeben wurde, so werden nach Eröffnen der Connection in dem entsprechenden Maskenfeld folgende Parameter angezeigt: Anwendungsname (z.B. $DIALOG), virt. Terminalname, Host, User-Id bei einer Connection zu $DIALOG.

Feld "**Program**"

Kurzbezeichnung des Verfahrens. Die Kurzbezeichnung erscheint in der Spalte "Verfahren" der Menue-Maske. Es dürfen in einem Menue keine gleichen Kurzbezeichnungen existieren. Eingegebene Kleinbuchstaben werden in Großbuchstaben umgesetzt.

Feld "**Connection-number or space**"

n 0 <= *n* <= 9 Feste Nummer der Connection, unter der das Verfahren eröffnet werden soll.

⊔ Blank: Das Menue-System sucht sich zur Eröffnung des Verfahrens die nächste freie Connection. Es werden hierbei jedoch die Nummern der Verfahren freigehalten, für die eine feste Connection-Nummer vergeben wurde (siehe oben). Die Nummer der ermittelten Connection wird nach erfolgreichem Eröffnen in die #-Spalte der Menue-Maske eingetragen.

Feld "**Break (B) allowed (Y/N)**"

B Die Eingabe 'B' oder 'K' bei dem Verfahren bewirkt die Auslösung der K2-Taste für die betreffende Connection.

⊔ Blank: Eingabe 'B' wird ignoriert und bewirkt keine Auslösung der K2-Taste für die Connection (notwendig für UTM-Verfahren).

Feld "**Plus-Module for P-keys**"

Optionaler Name eines PLUS-Moduls für eine spezielle P-Tastenbelegung zu dem Verfahren. Der Modul muß in der PLUS.MAPLIB oder in einer mit /FILE *datei*,LINK= MAPLIB zugewiesenen LMS-Bibliothek enthalten sein.

In bestimmten Fällen kann es notwendig sein, das automatische Umladen der P-Tasten beim Wechsel der Connection zu unterdrücken. Falls sich der Benutzer in einer Connection befindet und er das Umladen der P-Tasten beim Verlassen der Connection verhindern möchte, so ist das Kommando ^PL OFF einzugeben. Das automatische Umladen der P-Tasten kann jederzeit wieder eingeschaltet werden mit dem Kommando ^PL ON.

Feld "**Commands after Open-Connection**"

In diesem Bereich können bis zu 12 Eingaben festgelegt werden, die als automatische Startup-Eingaben nach dem erfolgreichen Eröffnen der Connection an den Partner

gesendet werden. Eingegebene Kleinbuchstaben werden in Großbuchstaben umgesetzt.

Als Startup-Sätze können auch beliebige Anweisungen aus der CFS-Prozedursprache (*-Anweisungen) angegeben werden. Auch CFS-interne Prozedurvariable können verwendet werden. Für eine ausführliche Beschreibung siehe Seite 485 und folgende (CFS-Prozedursprache).

Beispiele:

`KDCSIGN ....`
Anmeldung an ein UTM-Verfahren

`/DO prozedur,(...)`
Aufruf einer DO-Prozedur nach Anschluß an $DIALOG.

`user-id,C'passw'`
Eingabe der Verbindungsnachricht bei Anschluß an $CONSOLE.

`RES datei`
Ablaufenlassen eines mitgeschnittenen Eingangsdialogs.

Editieren mehrerer Zeilen im Feld "Commands after Open-Connection"

Zum Zwecke des Löschens, Einfügens und Kopierens bestehender Zeilen stehen besondere Editierkommandos zur Verfügung. Diese sind stets am Beginn der gewünschten Zeile einzutragen.

,c Copy. Die bezeichnete Zeile dient als Quelle für einen Kopiervorgang. Der durch ',c' überschriebene Zeileninhalt bleibt unverändert.

,m Move. Die bezeichnete Zeile wird verschoben. Das Ziel des Move-Vorgangs wird durch ',a', ',b' bzw. durch ',o' definiert. Der durch ',m' überschriebene Zeileninhalt bleibt unverändert.

,a After. Nach der bezeichneten Zeile wird die mit ',c' bzw. ',m' markierte Zeile eingefügt. Der durch ',a' überschriebene Zeileninhalt bleibt unverändert.

,b Before. Vor der bezeichneten Zeile wird die mit ',c' bzw. ',m' markierte Zeile eingefügt. Der durch ',b' überschriebene Zeileninhalt bleibt unverändert.

,o Over. Die bezeichnete Zeile wird mit dem Inhalt der mit ',c' bzw. ',m' markierten Zeile überschrieben.

,d Delete. Die bezeichnete Zeile wird gelöscht.

,i Insert. Vor der bezeichneten Zeile wird eine Leerzeile eingefügt.

Benutzerspezifische Gestaltung der Menue-Maske

Die Menue-Maske besitzt standardmäßig das Aussehen wie auf Seite 385 dargestellt. Falls gewünscht, kann der Systemverwalter oder jeder einzelne Benutzer ein spezielles, vom Standard abweichendes Layout der Menue-Maske definieren. Das gewünschte Maskenlayout wird in einer Formatdatei hinterlegt. Die Formatdatei muß FCBTYPE=SAM besitzen. CFS sucht in folgender Reihenfolge nach einer Formatdatei:

1) Existiert ein File-Kommando mit LINK=MENUEFOR ?

2) Mit dem im LOGON-Kommando angegebenen Job-Namen (/.*name* LOGON ...) wird nach einer Datei CFS.MENUE.FORMAT.*name* gesucht.

3) Wurde beim LOGON-Kommando kein Jobname angegeben oder existiert die Datei CFS.MENUE.FORMAT.*name* nicht, so wird unter der eigenen Benutzerkennung eine Datei CFS.FORMAT gesucht.

4) Existiert keine der oben genannten Dateien, so sucht CFS als letztes nach einer Datei $*user-id*.CFS.MENUE.FORMAT . $*user-id* ist die Benutzerkennung, unter der das Programm CFS gespeichert ist.

Der im folgenden dargestellte Inhalt der Formatdatei definiert das Standardlayout der Menue-Maske.

```
%DATG      %TIME              Host: %HOST          TSN: %TSN    Station: %TERM

                          $H  Application Menue  $R

  %Z  $U#$R $UHost$R        $UApplication$R  $UText$R
$UStatus$R
--------------------------------------------------------------------------------
  %a  $H%A$R%RA          %VA         %TA                                   %SA
  %b  $H%B$R%RB          %VB         %TB                                   %SB
  %c  $H%C$R%RC          %VC         %TC                                   %SC
  %d  $H%D$R%RD          %VD         %TD                                   %SD
  %e  $H%E$R%RE          %VE         %TE                                   %SE
  %f  $H%F$R%RF          %VF         %TF                                   %SF
  %g  $H%G$R%RG          %VG         %TG                                   %SG
  %h  $H%H$R%RH          %VH         %TH                                   %SH
  %i  $H%I$R%RI          %VI         %TI                                   %SI
  %j  $H%J$R%RJ          %VJ         %TJ                                   %SJ
  %k  $H%K$R             CFS         Return to CFS
  %l  $H%L$R             END         Terminate all applications
--------------------------------------------------------------------------------
        Password:   >%PASS      <            Lock: >%LOCK   <

$UMenue-File used    : %MENUE                                                $R
select application : X = activate, A = abort, B/K = break (K2), M = modify
%MESS
```

Die Formatdatei enthält neben konstanten Texten Variable als Platzhalter für bestimmte Maskenfelder, sowie Steuerfolgen zur Festlegung von Anzeigeeigenschaften.

Variable als Platzhalter für Maskenfelder

Variable beginnen stets mit dem Zeichen % und können an beliebiger Stelle in der Maske eingefügt werden. Variable benötigen in der Maske soviele Stellen, wie deren Länge (siehe unten) ausmacht. Es folgt eine Zusammenstellung aller möglichen Variablen für die Formatdatei.

Variable für **Eingabefelder**. Diese Felder werden in der Maske hell dargestellt.

%Z	Globales Eingabefeld der Menue-Maske. Länge = 1 + 1 geschützes Leerzeichen.
%a	Eingabefeld für Verfahren 0. Länge = 1 + 1 geschützes Leerzeichen.
%b	Eingabefeld für Verfahren 1. Länge = 1 + 1 geschützes Leerzeichen.
...	
%j	Eingabefeld für Verfahren 9. Länge = 1 + 1 geschützes Leerzeichen.
%k	Eingabefeld für Verfahren CFS (Rückkehr nach CFS). Länge = 1 + 1 geschützes Leerzeichen.
%l	Eingabefeld für Verfahren ENDE (Beendigung aller Verfahren). Länge = 1 + 1 geschützes Leerzeichen.
%PASS	Eingabefeld für das LOGON-Passwort. Länge = 11.
%LOCK	Eingabefeld für die LOCK-Funktion zum zeitweiligen Sperren des Bildschirms. Länge = 8.

Variable für **Ausgabefelder**. Diese Felder werden in der Maske standardmäßig halbhell dargestellt, sofern nicht durch besondere Angaben eine andere Darstellungsart gewählt wurde.

%A	Connection-Nummer für Verfahren 0. Länge = 2.
%RA	Host für Verfahren 0. Länge = 8.
%VA	Kurzbezeichnung des Verfahrens 0. Länge = 8.
%TA	Beschreibungstext zu Verfahren 0. Länge = 35.
%SA	Statusinformation zu Verfahren 0. Länge = 3.
...	
%J	Connection-Nummer für Verfahren 9. Länge = 2.
%RJ	Host für Verfahren 9. Länge = 8.
%VJ	Kurzbezeichnung des Verfahrens 9. Länge = 8.
%TJ	Beschreibungstext zu Verfahren 9. Länge = 35.
%SJ	Statusinformation zu Verfahren 9. Länge = 3.
%K	'C ' für Verfahren CFS (Rückkehr nach CFS). Länge = 2.
%L	'E ' für Verfahren ENDE (Beendigung aller Verfahren). Länge = 2.
%TSN	TSN des Prozesses, unter dem CFS aufgerufen wurde. Länge = 4.
%DATG	aktuelles Datum in der Form tt.mm.jj. Länge = 8.

%DATE	aktuelles Datum in der Form mm.dd.jj. Länge = 8.
%TIME	aktuelle Uhrzeit in der Form hh:mm:ss. Länge = 8.
%HOST	Name des Hostrechners, unter dem CFS aufgerufen wurde. Länge = 8.
%USER	Benutzerkennung, unter der CFS aufgerufen wurde. Länge = 8.
%MENUE	Name der verwendeten Menue-Datei. Länge = 40.
%TERM	Name des Terminals, an dem CFS aufgerufen wurde. Länge = 40.
%MESS	Meldungstext (Hinweis auf Help-Funktion, Grund für Verbindungsabbruch usw.). Länge = 70.

Steuerzeichen zur Veränderung der Anzeigeeigenschaften von Ausgabefeldern. Standardmäßig werden Ausgabefelder halbhell dargestellt. Die im folgenden aufgeführten zwei Byte langen und mit dem Zeichen $ beginnenden Steuerfolgen schalten die Anzeigeeigenschaften um, belegen in der Maske jedoch keinen Platz. Die Maskendefinition für eine Zeile verlängert sich für jede Steuerfolge um zwei Byte.

$B	Hell und unterstrichen
$H	Hell
$U	Unterstrichen
$R	Reset. Halbhell, nicht unterstrichen

Meldungen bei **fehlerhafter Maskendefinition**:

000001	ungültiger Name einer %-Variable
000002	Eingabevariable (%a, ..., %l, %PASS, %LOCK) mehrfach angegeben
000003	Eingabevariable (%a, ..., %l, %PASS, %LOCK) fehlt
000011	ungültige Anzeigesteuerfolge ($B, $H, $U, $R)

Hinweis: In der ausgelieferten Bibliothek CFS.S.LMSLIB sind die Formatdefinitionen der Standardmaske sowie eines alternativen Maskenlayouts als Elemente X/CFS. MENUE.FORMAT.2 bzw. 1 gespeichert. Die durch CFS.MENUE.FORMAT.1 definierte alternative Maske besitzt folgendes Aussehen:

```
*-----------------------------------------------------------------------------*
!   CFS - Dialogsteuerung             Host: HOST1     TSN: xxxx    dd.mm.yy  hh:mm:ss !
!-----------------------------------------------------------------------------*
! Menue : xxxxxxxxxxxxxxxxxx                              Station : ssssssss !
! X = Verfahren auswaehlen, A = Abbruch, B/K = K2-Taste, M = Eintrag aendern   !
! oder Verfahrens-Nummer aus #-Spalte angeben                                  !
*-----------------------------------------------------------------------------*
!   ! # Rechner     !  Verfahren ! Beschreibung                       ! Status !
*-----------------------------------------------------------------------------*
!   !   HOST1       !  UTMTEST   !  UTM-Anwendung Lager (TEST)         !  OFF   !
!   ! 0 HOST2       !  CONS-2    !  UCON-Verbindung Rechner 2          !  ON    !
!   !   HOST1       !  ODR1      !  $DIALOG mit User-ID $ODR1          !  OFF   !
!   ! 1 HOST3       !  SAP-P     !  SAP-Verbindung (Produktiv)         !  ON    !
!   ! 4 HOST1       !  SAP-T1    !  SAP-Verbindung (Test 1)            !  OFF   !
!   ! 5 HOST2       !  SAP-T2    !  SAP-Verbindung (Test 2)            !  OFF   !
!   !               !            !                                     !        !
!   !               !            !                                     !        !
!   !               !            !                                     !        !
!   !               !            !                                     !        !
!   ! C             !  CFS       !  Programm CFS                       !        !
!   ! E             !  ENDE      !  Beenden aller Verfahren            !        !
*-----------------------------------------------------------------------------*
! Passwort : >...........<          Lock: >........<                           !
* for Help type "?" at any Input-Field                                         *
```

11. Job-Report System

Mit dem Job-Report System (JRS) von CFS können Enter-Aufträge und DO-Prozeduren gestartet und überwacht werden.

Gegenüber den BS2000-Kommandos /DO und /ENTER bietet JRS folgende Erweiterungen:

1) Neben Enter-Dateien können auch DO-Prozeduren im Enter-Modus gestartet werden. Bei DO-Prozeduren wird von JRS eine Enter-Datei erzeugt, die den gewünschten DO-Aufruf enthält.

2) Wenn eine DO-Prozedur im Enter-Modus gestartet wird, werden die DO-Parameter von JRS analysiert und können im formatgesteuerten Prompting vom Benutzer vervollständigt werden (Parameter-Maske).

3) JRS benachrichtigt den Benutzer über die Beendigung der Enter-Aufträge. Dabei wird neben allgemeinen Informationen über den Job wahlweise die SYSOUT-, SYSLST- oder eine beliebige andere Datei angezeigt. Wenn das Programm CFS zum Zeitpunkt der Beendigung des Enter-Auftrags nicht mehr geladen ist, so wird die Beendigung des Auftrags angezeigt, sobald CFS zum ersten Mal wieder unter der gleichen Benutzerkennung und Abrechnungsnummer aufgerufen wird. Die Benachrichtigung über einen zum Ende gekommenen Enter-Auftrag erfolgt unabhängig davon, ob der Benutzer gerade im Grundprozeß von CFS oder in einer Connection arbeitet.

4) CFS bietet über ein eigenes Status-Kommando (STAJ) die Möglichkeit, Informationen über alle durch JRS überwachten Enter- und File-Transfer-Aufträge abzurufen.

5) Wenn in einer Connection ein /DO-Kommando abgesetzt wird, so übernimmt JRS automatisch die DO-Parameter Analyse. Bei unbesetzten Parametern wird die Parameter-Maske ausgegeben. Der Benutzer erhält dabei eine formatierte Liste aller Parameter der Prozedur und kann (muß aber nicht) noch fehlende Parameterwerte ergänzen. Falls die Connection in einer anderen Benutzerkennung oder auf einem anderen Host-Rechner eröffnet wurde, so kann diese Funktion nicht genutzt werden.

   Falls die Connection zu einem Dialogprozeß in einer anderen Benutzerkennung eröffnet wurde, so kann JRS bei /DO-Kommandos genutzt werden, falls die angesprochene DO-Prozedur shareable ist.

6) JRS bietet auch die Möglichkeit, Enter-Aufträge zu überwachen, die nicht aus CFS mit dem Kommando E gestartet wurden. Der Benutzer kann dabei auch über beliebige Steps innerhalb eines Enter-Auftrags informiert werden. Es können auch beliebige andere, an eine Jobvariable gebundene Events überwacht werden.

7) Mit JRS können auch File-Transfer Vorgänge überwacht werden, die mit dem FT-Kommando von CFS gestartet wurden.

Die Job-Report Funktion kann nur benutzt werden, wenn im BS2000-System Job-Variable installiert sind.

Enter-Datei starten

E { *enter-datei* | *bibl*(*element*) } [/*enter-params* | , ?] [, JRF=*jr-file*] [, JRINIT=Y]

DO-Prozedur im Enter starten

E[N[R],] { *do-prozedur* | *bibl*(*element*) } [, *do-params* | , ?] [/ [*enter-params*] [, JRF=*jr-file*] [, JN=*jobname*]] [, JRINIT=Y]

Enter- / DO-Prozedur mit vorher definierten Parametern starten

E[N[R],] %*name* | %? [, *jrsave*]

Die Parameter müssen zu einem früheren Zeitpunkt mit dem Kommando SP (Save Params) gesichert worden sein. Das SP-Kommando kann nur im Kommandofeld der Parameter-Maske eingegeben werden.

Letzte Enter- / DO-Prozedur mit gleichen Parametern anzeigen

E Das E-Kommando ohne irgendwelche Operanden bewirkt, daß die Parameter-Maske der zuletzt gestarteten DO-/Enter-Prozedur am Bildschirm angezeigt wird und wieder ausgeführt werden kann.

Parameter:

N Der Zusatz N im E-Kommando bewirkt, daß die Maske zur Vervollständigung der Parameter für eine DO-Prozedur (siehe Seite 400) nicht ausgegeben wird. Alle gewünschten DO-Parameter müssen bereits im E-Kommando angegeben werden.

NR Der Zusatz NR im E-Kommando bewirkt wie bei N (s.o.), daß die Maske zur Vervollständigung der Parameter für eine DO-Prozedur nicht ausgegeben wird. Alle gewünschten DO-Parameter müssen im E-Kommando angegeben werden. Im Gegensatz zum Parameter N unternimmt CFS auch keinen Versuch, die DO-Prozedur zu lesen.

enter-datei Name der zu startenden Enter-Datei. Die Enter-Datei muß mit einem LOGON-Kommando beginnen und mit einem LOGOFF-Kommando enden bzw. mit den entsprechenden SDF-Kommandos. Falls bei Beendigung des Enter-Auftrags eine Ergebnisliste (SYSOUT/SYSLST) angezeigt werden soll, so muß in der Enter-Datei ein entsprechendes SYSFILE SYSOUT-/SYSFILE SYSLST-Kommando bzw. das entsprechende SDF-Kommando enthalten sein.

do-prozedur Name der im Enter zu startenden DO-Prozedur. Die DO-Prozedur muß mit einer PROC-Anweisung beginnen und mit einer ENDP-Anweisung enden bzw. mit den entsprechenden SDF-Kommandos. In die von JRS erzeugte Enter-Datei, die einen CALL auf die DO-Prozedur enthält, wird ein SYSFILE SYSOUT-/SYSFILE SYSLST-Kommando eingefügt.

bibl(*element*) Die Enter-Datei/DO-Prozedur ist im angegebenen Bibliothekselement gespeichert. Bei PLAM-Bibliotheken muß das Bibliothekselement unter dem Elementtyp J abgelegt sein.

%*name* Name des Param-Sets in der JRSAVE-Datei. Die mit SP %*name* gesicherte Parameter-Maske wird zum Enter angeboten. Mit %*name* wird ein Bezug sowohl auf eine

bestimmte DO-Prozedur, als auch auf eine vordefinierte Belegung der DO-Parameter hergestellt (näheres siehe Beschreibung des Kommandos SP auf Seite **403**).

%? Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

jrsave zu verwendende JRSAVE-Datei. Der Name sollte nur angegeben werden, falls der Param-Set nicht in der Standard JRSAVE-Datei gespeichert ist. Die Regel, nach der Name der JRSAVE-Datei gebildet wird, ist auf Seite **564** unter CFS. JRSAVE beschrieben.

do-params Parameter für die zu startende DO-Prozedur. Die Angabe von *do-params* kann entfallen, wenn die zu startende DO-Prozedur keine Parameter besitzt, wenn alle Parameter mit Standardwerten versorgt sind oder wenn der Benutzer die DO-Parameter-Maske für die zu startende Prozedur wünscht. DO-Parameter können nicht zusammen mit %*name* angegeben werden.
Im Standardfall werden die angegebenen Parameterwerte unabhängig von der Eingabe, in Großbuchstaben an die BS2000-Prozedur weitergereicht.
Ausnahme: L'*wert*' bzw. PARAM=L'*wert*' In diesem Fall wird L vor der Übergabe an das BS2000 vom Job-Report System entfernt. Der Parameterwert wird jedoch unverändert in der angegebenen Schreibweise an das BS2000 übergeben.

? Das "?" anstelle der DO-Parameter bewirkt, daß die Parameter-Maske auf jeden Fall ausgegeben wird - auch wenn alle Parameter befriedigt sind oder wenn in der DO-Prozedur kein Parameter vorkommt.

/*enter-params* BS2000-Enter-Parameter. Es können alle Parameter des BS2000 Enter-Kommandos angegeben werden, z.B. TIME=*nn*,JOB-CLASS=*xxxx*,PRIORITY=*nnn*

Falls eine echte Enter-Datei über das Job Report System gestartet wird, so kann im Kommando E *enter-datei*/*enter-params* der Schrägstrich auch durch ein Komma ersetzt werden: E *enter-datei*,*enter-params* (Kompatibilität zum BS2000 Enter-Kommando).

Die Enter-Parameter können auch mit dem Kommando JREP oder in der JRP-Maske vorbelegt werden. Die vorbelegten Enter-Parameter werden durch die Enter-Parameter im E-Kommando ergänzt oder überschrieben (gleiche Schlüsselworte).

? Durch "?" anstelle der Enter-Parameter wird eine Maske ausgegeben, in der die Standard Enter-Parameter (siehe Kommando JREP, Seite **410**) aufgeführt sind und verändert bzw. ergänzt werden können.

JRF= SYSOUT | SYSLST
{ *datei* | *LIB(*bibl*,*element*[(*vers*)] [,*typ*] } , SYSOUT | SYSLST | <u>JRFILE</u>]
Falls bei einen Bibliothekselement (*LIB ...) die Typ-Angabe weggelassen wurde, so wird das Element unter dem Typ P gespeichert.

Mit dem wahlweisen Parameter JRF wird die bei Jobende anzuzeigende Datei bestimmt. Für eine genauere Beschreibung des Parameters JRF siehe Kommando JRF Seite **411**.

Normalerweise wird die SYSOUT-Datei des gestarteten Jobs angezeigt. Wenn nichts angegeben wird, wird der Name der SYSOUT-Datei oder einer anderen JRFILE aus der Enter-Datei oder der DO-Prozedur ermittelt. JRS durchsucht die Prozedur-Datei nach den Anweisungen (SYSFILE, ASSIGN-SYSOUT, ASSIGN-SYSLST oder REMARK JRF=).

Wenn eine DO-Prozedur als Enter-Datei gestartet werden soll, wird von JRS, je nach JRF-Option, folgende Datei erzeugt:

CFS.SYSOUT.*tsn.hhmmss* oder
CFS.SYSLST.*tsn.hhmmss*

Bei einem SYSLST-Protokoll wird in der Enter-Datei das Kommando OPTION MSG=FH eingefügt.

Je nach der Option im JRSH-Kommando bzw. der JRP-Maske wird die SYSOUT-/ SYSLST-Datei mit dem Attribut SHARE=YES/NO versehen (Standard: SHARE=NO). Die Option SHARE im JRSH-Kommando hat den Vorteil, daß die Kommandos JR und STAJ auch für fremde User-ID's zur Verfügung stehen.

Der JRF-Parameter kann auch über eine REMARK-Anweisung in der Enter-Datei angegeben werden (/REMARK JRF=...).

JN=*jobname* ein- bis achtstelliger Jobname für den zu startenden Enter-Auftrag.

JRINIT=Y Mit dieser Angabe kann das folgende Standardverhalten des Job-Report Systems geändert werden: Annahme TSN x und y sind unter der gleichen kennung angeloggt, haben CFS aufgerufen und haben das Job-Report System mit der gleichen JR-ID aktiviert. Unter TSN x wird mit Hilfe des E-Kommandos ein Enter-Job gestartet. CFS unter TSN y wird nicht von der Beendigung des Jobs informiert. Im STAJ-Kommando unter TSN y wird der Job mit Status NJR angezeigt.

JRINIT=Y bewirkt, daß auch unter TSN y der Enter-Job aktiv überwacht wird. Voraussetzung hierfür ist jedoch, daß unter der Benutzerkennung eine Jobvariable CFS.JR.*jrid*.JRINIT eingerichtet wurde. *jrid* ist die Job-Report-ID, unter der die Enter-Aufträge überwacht werden.

Prüfung der Parameter für DO-Prozedur

Wird eine DO-Prozedur mit dem E-Kommando gestartet, so wird dies von JRS anhand der PROC-Anweisung in der zu startenden Datei erkannt. Es werden dann zusätzliche Plausibilitätsprüfungen vorgenommen. JRS erkennt z.B., wenn ein Parameter (Stellungs- oder Schlüsselwortparameter) noch nicht versorgt ist (keine Angabe über *do-params* im E-Kommando und kein Standardwert in der PROC-Anweisung definiert). In diesem Fall erhält der Benutzer eine Maske mit einer Auflistung aller Parameter der DO-Prozedur zusammen mit den zu diesem Zeitpunkt bereits vorliegenden Parameterwerten. Die Gestaltung der Parametermaske kann für jeden einzelnen Parameter, d.h. in jeder Zeile durch besondere Steueranweisungen variiert werden. Die i.f. beschriebenen Steueranweisungen sind mit " einzuleiten und zu beenden. Die einzelnen Anweisungen können in einzeln oder kombiniert und in beliebiger Reihenfolge angegeben werden.

Kommentierung der Parameter

Neben den Parameternamen werden auch Beschreibungen zu den Parametern in der Maske angezeigt (bis zu einer Länge von 40 Bytes).

Beispiel: Ist in der PROC-Anweisung eine Zeile der folgenden Art enthalten: /&DATEI=TEST, "Name der zu verarbeitenden Datei"-
so wird dieser DO-Parameter in der Maske in folgender Form angezeigt:

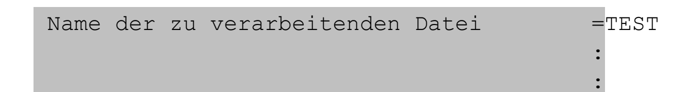
```
 DATEI Name der zu verarbeitenden Datei  =TEST
```

: Beginnt der Kommentar mit dem Zeichen ":" , so wird nur der Kommentar ohne den Parameternamen angezeigt.
Beispiel: Ist in der PROC-Anweisung eine Zeile der folgenden Art enthalten:
/&DATEI, ":Name der zu verarbeitenden Datei"-
so wird dieser Parameter in der Maske in folgender Form angezeigt:

```
Name der zu verarbeitenden Datei        :
```

Anzahl der Eingabezeilen für Parameter

#*n* Durch #*n* (*n* < 5) können für einen Parameter bis zu vier zusätzliche Eingabezeilen zu je 49 Byte reserviert werden. #*n* muß nach dem ersten " bzw. nach ": angegeben werden. Standardmäßig wird in der Bildschirmmaske für jeden Prozedurparameter eine Zeile reserviert. Dies bedeutet, daß Parameterwerte nur in der Länge von 49 Byte angegeben werden können.
Beispiel: Ist in der PROC-Anweisung eine Zeile der folgenden Art enthalten:
/&DATEI=TEST, ":#2Name der zu verarbeitenden Datei"-
so wird dieser Parameter in einem dreizeiligen Eingabefeld angefordert:



Länge des Eingabefeldes für Parameter

#L*n* Durch #L*n* (*n* < 246) kann die Länge des Eingabefeldes für einen Parameter festgelegt werden. Falls eine Länge > 49 angegeben wird, werden die notwendigen zusätzlichen Zeilen automatisch reserviert.
Beispiel: Ist in der PROC-Anweisung eine Zeile der folgenden Art enthalten:
/&DATEI=TEST, ":#L20Optionen" -
so wird in der Params-Maske nur ein überschreibbares Feld in der Länge 20 ausgegeben. Eine längere Eingabe ist dann nicht möglich.

Numerischer Parameterwert

#N Durch #N wird kann im Eingabefeld für den Parameter nur ein numerischer Wert angegeben werden.

Dunkelgesteuertes Eingabefeld für Parameter

#D Durch #D kann das Eingabefeld dunkel gesteuert werden.
Beispiel: /&PASSW=, ":#DLogon-Passwort fuer Kennung"-

Eingabewert für Parameter nicht in Großbuchstaben umwandeln

#U Durch #U wird bestimmt, daß der in dem Maskenfeld eingegebene Parameterwert unverändert, d.h. ohne Umwandlung in Großbuchstaben an die Prozedur weitergegeben wird.

Parameter nicht in Maske ausgeben

#- Durch #- wird veranlaßt, daß der Parameter in der Maske nicht ausgegeben wird.
Beispiel: /&PASSW=C'ABC', ":#- Logon-Passwort nicht im Maske"-

Hinweise:

Durch die Kommentare zu den Prozedurparametern ist es möglich, die Parameter-Maske übersichtlicher und aussagefähiger zu gestalten. Die o. a.. Optionen (#*n*/#D/#L*n*/#-/#U/:) können in beliebiger Kombination und Reihenfolge angegeben werden. Sie müssen aber immer am Anfang der Bemerkung nach dem Zeichen " stehen.

In der Parameter-Maske können Prozedurparameter, die im DO-Kommando nicht oder nicht vollständig angegeben wurden, nachgetragen werden. Durch die Eingabe E im Kommandofeld der Parameter-Maske wird der Enter gestartet.

Durch Drücken der K1-Taste in der Parameter-Maske wird das E-Kommando annulliert, der Enter wird nicht gestartet.

Durch Eingabe von "?" an der Stelle der DO- oder Enter-Parameter wird die Parameter-Maske angefordert. Das Kommando E *do-proc*,? bzw. E *bibl*(*element*),? kann auch verwendet werden, wenn sich der Benutzer nur einen Überblick über die vorhandenen Stellungs- und Schlüsselwortparameter und deren Vorbelegung verschaffen möchte. Damit kein Enter-Auftrag gestartet wird, muß die Parameter-Maske in diesem Fall mit der K1-Taste verlassen werden. Der vorbelegte oder vom Benutzer modifizierte Inhalt der Parameter-Maske kann mit dem Kommando SP für eine spätere Wiederverwendung (E %*name*) gespeichert werden.

Beispiel:

Die DO-Prozedur DO.COB besitzt folgende PROC-Anweisung:

```
/PROC N,(&MODUL,                                                    -
/         &COLNAS=N,                "#L1"                           -
/         &COLLIST=Y,               "#L1"                           -
/         &COLCOB=Y,                "#L1"                           -
/         &COBOLCOMPILER=Y,         "#L1"                           -
/         &COLNUMC=Y,               "#L1"                           -
/         &COBLIB=$COPLIB,          "Cobol-COPY-Lib"                -
/         &COBLIB1=$COPLIB,                                         -
/         &COBLIB2=$COPLIB,                                         -
/         &COPYCLIB1=$COPLIB,       "Columbus-COPY-Lib"             -
/         &COPYCLIB2=$COPLIB,                                       -
/         &MODLIB=MODLIB,                                           -
/         &COLLISTLISTE=,                                           -
/         ),SUBDTA=&
```

Aufgrund des Kommandos
`E DO.COB`
wird die folgende Parameter-Maske ausgegeben:

```
   dd.mm.yy   hh:mm:ss          J O B - R E P O R T                TSN: ....
   COMMAND :E DO.COB

                                 D O - P A R A M S
   MODUL                        :
   COLNAS                       =N
   COLLIST                      =Y
   COLCOB                       =Y
   COBOLCOMPILER                =Y
   COLNUMC                      =Y
   COBLIB                       =$COPLIB
   COBLIB1                      =$COPLIB
   COBLIB2                      =$COPLIB
   COPYCLIB1                    =$COPLIB
   COPYCLIB2                    =$COPLIB
   MODLIB                       =MODLIB
   COLLISTLISTE                 =

                             E N T E R - P A R A M S


   JR-Filename : $TT08.CFS.JR.TEST.SYSOUT.02BC.132505                    ,SYSOUT


    + / - / -- / ++ / E(nter) / A(bort) (K1) / "/cmd"        -      FOR HELP: ?
```

In der Parameter-Maske sind die Stellungsparameter der PROC-Anweisung mit **":"** und die Schlüsselwortparameter der PROC-Anweisung mit **"="** gekennzeichnet.

Kommandos in der Parameter-Maske

+ | **-** Eine Bildschirmseite vorwärts/zurück blättern.

++ | **--** Letzte/erste Bildschirmseite anzeigen.
Neben "+" führt auch das leere Kommandofeld (abgesendet mit ENTER) auf den nächsten Bildschirm.

A Parameteranalyse abbrechen (gleiche Wirkung wie Betätigung der K1-Taste).

/ CFS unterbrechen und in Betriebssystem-Modus wechseln.

/cmd BS2000-Kommando ausführen.

STAJ Status der von JRS überwachten Aufträge ausgeben. Näheres siehe Seite **404**.

JR | **NJR** JRS aktivieren / deaktivieren. Näheres siehe Seite **410**.

JRP Globale JRS-Parameter setzen. Näheres siehe Seite **414**.

SP %*name* [,*jrsave*] Save Parameters. Speichern der Parameter-Maske.
Mit diesem Kommando wird die vom Benutzer vervollständigte Parameter-Maske mit einem festgelegten Namen in der Datei CFS.JRSAVE [.*jobname*] bzw. in einer mit /FILE *datei*,LINK=JRSAVE zugewiesenen Datei gespeichert, sofern im SP-Kommando der Zusatz *jrsave* nicht angegeben wurde. Die genaue Regel, nach der der Name der JRSAVE-Datei gebildet wird, ist auf Seite **564** unter CFS.JRSAVE beschrieben.

%*name* Im Param-Set %*name* werden die Belegung der DO-Parameter, der Name der verwendeten DO-Prozedur, sowie evtl. angegebene Enter-Parameter gespeichert. Mit dem Kommando /FILE *datei*,LINK=JRSAVE kann die verwendete Parameter-Datei auch vom Benutzer selbst zugewiesen werden.

jrsave Name der JRSAVE-Datei. Der Name sollte nur dann angegeben werden, wenn die Parameter in eine andere als die von CFS standardmäßig verwendete JRSAVE-Datei (siehe oben) gesichert werden.

Durch das Kommando E %*name* wird die unter dem Namen gesicherte Parameter-Maske wieder zum Enter angeboten.

In Connections (Nebenprozessen), die unter der eigenen Kennung eröffnet wurden, wird durch das Kommando DO *%name* ebenfalls in die Parameter-Maske verzweigt. Mit DO %*name* kann somit eine früher bestimmte DO-Prozedur und deren Parameterbelegung für den Ablauf im Dialog aktiviert werden.

In Connections ebenfalls zulässig ist das Kommando 'DO '. Hiermit wird die zuletzt gestartete DO-Prozedur mit den zuletzt angegebenen Parametern wieder aktiviert (Anzeige der entsprechenden Parameter-Maske).

E Enter-Auftrag starten (oder K3-Taste). Das Kommandofeld der Parameter-Maske ist standardmäßig mit dem Kommando E vorbelegt, indem dort das volle vom Benutzer angegebene E-Kommando wiederholt wird: E *do-prozedur*,...

Das Kommando E ist auch beim Start von DO-Prozeduren in Connections einzugeben, falls sich die Maske der DO-Parameter über mehrere Bildschirme erstreckt und der Benutzer nicht bis zum letzten Bildschirm blättern will.

Status der gestarteten Enter-Aufträge anzeigen

Das STAJ-Kommando kann in der Kommandozeile von CFS, in der Job-Report Parameter Maske (E-Kommando), sowie in der Job-Report Maske (Ergebnis-SYSOUT Listing) angegeben werden.

STAJ [$*userid* | *jr-id* | $*user-id* . *jr-id*]

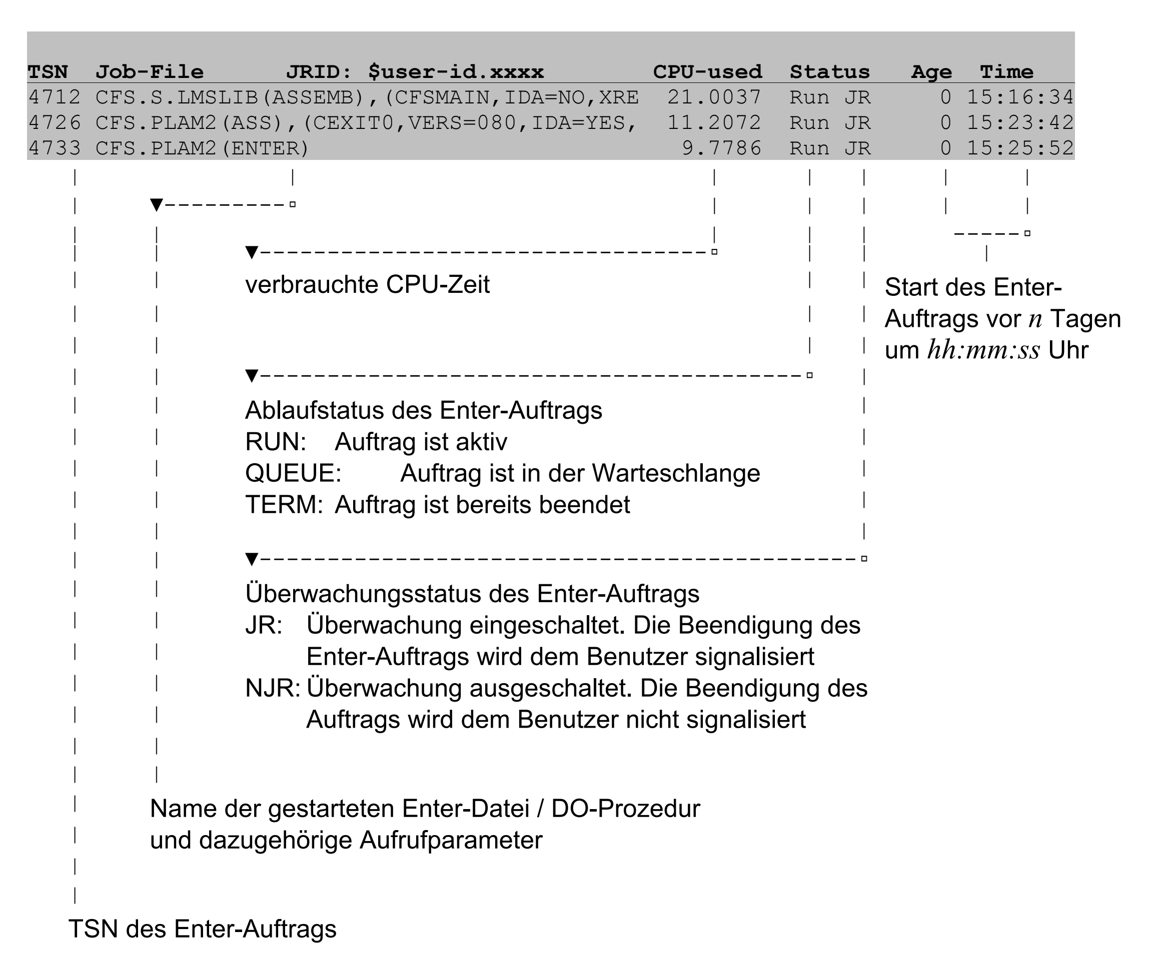
jr-id Diese wahlweise Angabe bewirkt, daß der Status lediglich für diejenigen Enter-Aufträge ausgegeben wird, die unter der entsprechenden Job-Report ID gestartet wurden. Siehe Kommando JRID Seite **413**.

Wenn kein Parameter im STAJ-Kommando angegeben wurde, werden alle Jobs angezeigt, die unter der Standard JR-ID (= *acct-nr* bzw. Jobname) gestartet wurden oder unter einer JR-ID, deren Überwachung mit dem JR-Kommando explizit aktiviert wurde.

Hinweis:

In besonderen Bildschirmmodi (TCHNG READ=U) bzw. bei Terminalemulationen oder unter OMNIS kann der Fall eintreten, daß der Bildschirm nach Ausgabe der STAJ-Informationen wieder gelöscht wird. In den meisten Fällen kann dieses Problem durch Setzen des Prozeßschalters 2 (/SETSW ON=2) behoben werden.

Ausgabeinformationen des STAJ-Kommandos:



TSN	Job-File JRID: $user-id.xxxx	CPU-used	Status	Age	Time
4712	CFS.S.LMSLIB(ASSEMB),(CFSMAIN,IDA=NO,XRE	21.0037	Run JR	0	15:16:34
4726	CFS.PLAM2(ASS),(CEXIT0,VERS=080,IDA=YES,	11.2072	Run JR	0	15:23:42
4733	CFS.PLAM2(ENTER)	9.7786	Run JR	0	15:25:52

Die Job-Report Maske

Die Job-Report Maske wird ausgegeben, sobald ein mit E ... gestarteter Enter beendet wurde. In der Job-Report Maske werden Statistikinformationen über den Enter-Auftrag sowie die JR-File (SYSOUT-Datei, SYSLST-Datei bzw. beliebige Datei) angezeigt.

Beispiel für die Job-Report Maske, wie sie nach Beendigung eines mit dem E-Kommando gestarteten Enter-Auftrags automatisch ausgegeben wird.

```
dd.mm.yy   hh:mm:ss          J O B - R E P O R T                TSN: ....
COMMAND :

Job-File     :      CFS.S.LMSLIB(ASSEMB),(CFSMAIN,IDA=NO,XREF=NOX   TSN: 4712
Job started :       today       at 15:16         JRID      : $ABC.CFS
    by TSN   :      4711
Termination :       normally    at 15:29         CPU-Time : 36.3953  sec.
SYSOUT-file :       $ABC.CFS.SYSOUT.4711.151631

- /STEP
- /DO CFS.S.LMSLIB(ASSEMB),(CFSMAIN,IDA=NO,XREF=NOXREF,)
- %  BLS0500 PROGRAMM 'ASSEMB', VERSION '300' VOM '87-09-01' WURDE GELADEN.
- GIVE ASSEMBLER OPTIONS !
- EOF REACHED ON SYSDTA; *END ASSUMED
- FLAGS IN 00000 STATEMENTS, 000 PRIVILEGED FLAGS, 000 MNOTES
- HIGHEST ERROR WEIGHT : -
- USER MACROLIBRARY :     CFS.MACLIB.USR
- 2ND USER MACROLIBRARY: $TSOS.MACROLIB
- 3RD USER MACROLIBRARY: $TSOS.SYSMAC.DCM
- SYSTEM MACROLIBRARY :  MACROLIB
- SOURCE LIBRARY  :      CFS.S.LMSLIB


  + / - / -- / ++ / e (K1) / k / n / njr (K3) / "/cmd"      -     FOR HELP: ?
```

Die Job-Report Maske enthält, ähnlich wie die Dateienliste von CFS, ein Kommandofeld. Hier kann mit einer Reihe von Kommandos im Listing geblättert und positioniert werden.

Eingaben im Kommandofeld der Job-Report-Maske

+ | **-** In der JR-File um einen Bildschirm weiter zum Ende / zum Anfang blättern (Standard: weiter in Richtung Ende blättern).

+*n* | **-***n* In der JR-File um *n* Sätze weiter zum Ende / zum Anfang blättern.

++ | **--** Zum Ende / zum Anfang der JR-File blättern.

DS | DL [DW] Display short / Display Long.
DS: Es werden nur die ersten 80 Bytes jedes Datensatzes des Sysout-Protokolls angezeigt. Es erfolgt kein Umbruch in die nächste Zeile.
DL (DW): Bei Datensätzen, die länger als 80 Bytes sind, werden die folgenden Daten in der nächsten Bildschirmzeile fortgesetzt (Standard).
Eine Änderung des DS/DL-Modus bleibt bis zum Ende der CFS-Sitzung bestehen.

S,'*string*' In der JR-File auf den angegebenen Suchbegriff positionieren. Wird der Begriff nicht gefunden, so wird auf den Anfang der JR-File positioniert. Vollständige Syntaxbeschreibung des Suche-Kommandos siehe Seite 309.

S Zum nächsten Auftreten der zuletzt definierten Suchzeichenfolge positionieren.

/cmd	BS2000-Kommando ausführen.
/	In den BS2000-Kommandomodus verzweigen.

E | **K1**

End + Erase SYSOUT + ERAJV monjv.

Der Vorgang "benachrichtige Benutzer über die Beendigung des Enter-Auftrags" ist damit abgeschlossen. Falls als JR-File das SYSOUT-Protokoll angezeigt wurde, so wird dieses gelöscht bzw. bei eingeschalteter ERT-Option, in eine temporäre Datei umkatalogisiert. Ein angezeigtes SYSLST-Protokoll bzw. eine andere JR-File wird nicht gelöscht. Für weitere Informationen zum ERT-Modus siehe Seite **474**. Falls das SYSOUT- bzw. SYSLST-Protokoll in einer PLAM-Bibliothek gespeichert ist, wird dieses bei K1 ebenfalls nicht gelöscht.

EN

Nur bei benutzerdefinierten Ereignissen (siehe Seite **408**):
Beenden des Job-Report (wie E bzw. K1-Taste). Die erneute Überwachung von Ereignissen durch diese Jobvariable wird nicht aktiviert. Standard: Die Überwachung wird aktiviert. Das Eintragen von 'N' in der Spalte 255 der überwachenden Jobvariablen durch den Benutzer bewirkt, daß auch bei Angabe des Kommandos E bzw. K1-Taste die Jobvariable nicht weiter überwacht wird.

Hinweis:

Falls der Benutzer vor Ausgabe der Job-Report Maske im Format-Modus des EDT eine Datei bearbeitet hat, so ist der ursprüngliche Bildschirminhalt nach Beendigung des Job-Report in der Regel zerstört. Der ursprüngliche EDT-Bildschirm kann durch Betätigung der K3-Taste wieder hergestellt werden.

EP [*x*]

End + Print. Neben den Funktionen des Kommandos E wird zusätzlich die angezeigte JR-File (SYSOUT, SYSLST oder beliebige Datei) ausgedruckt.

x — Für *x* kann eine beliebige Variante des Print Action-Codes angegeben werden.

Beispiele:

EPSE	End + Print Space=E,Erase.
EPDnS	End + Print on Device *n*, Space=E

Für eine vollständige Beschreibung aller Varianten des Action-Codes P siehe Seite 6-**201**.

K

End + Keep. Der Vorgang "benachrichtige Benutzer über die Beendigung des Enter-Auftrags" ist damit abgeschlossen (wie bei E...). Das SYSOUT-Protokoll wird jedoch nicht gelöscht.

KP [*x*]

End + Keep + Print. Neben den Funktionen des Kommandos K wird zusätzlich die angezeigte JR-File (SYSOUT, SYSLST oder beliebige Datei) ausgedruckt

x — siehe oben: Kommando EP*x*.

N

No action. Die Anzeige der Job-Report Maske für einen beendeten Enter-Auftrag wird abgebrochen. Der Vorgang "benachrichtige Benutzer über die Beendigung des Enter-Auftrags" ist damit jedoch noch nicht abgeschlossen.

Die Eingabe von N ist sinnvoll, wenn die Job-Report Maske einem CFS-Benutzer zugestellt wird, der den betreffenden Enter-Auftrag nicht gestartet hat oder wenn die Anzeige der JR-File im Moment nicht gewünscht wird. Bei einem neuen Aufruf von CFS bzw. nach Eingabe des Kommandos JR wird diese Job-Report Maske wieder angezeigt. Dies geschieht solange, bis ein Benutzer diese Job-Report Maske mit Kommando K, E oder mit der K1-Taste quittiert.

Die Anzeige der JR-File kann somit durch Eingabe des N-Kommandos in der Job-Report Maske unterbrochen und mit dem Kommando JR wieder fortgesetzt werden.

Hinweis:

In seltenen Fällen kann es vorkommen, daß beim Versuch, nach Beendigung eines Enter-Auftrags die JR-File in der Job-Report Maske anzuzeigen, der DMS-Fehler 0D9A auftritt. Diese Fehlersituation kann behoben werden durch Drücken der Taste ENTER oder durch Eingabe von N (No Action) und anschließendes Kommando JR.

NP [*x*] No action + Print. Neben den Funktionen des Kommandos N wird zusätzlich die angezeigte JR-File (SYSOUT, SYSLST oder beliebige Datei) ausgedruckt (analog Kommando EP, siehe oben).

x siehe oben: Kommando EP*x*.

NJR | **K3** No Job Report. Gleiche Wirkung wie N (No action), jedoch wird auch die Ausgabe der Job-Report Masken für alle weiteren bereits beendeten Enter-Aufträge unterdrückt.

STAJ [*jr-id*] Status der mit dem E-Kommando gestarteten Enter-Aufträge anzeigen.

Als weitere Kommandos sind in der Job-Report-Maske erlaubt:

JRP / JRF / JRIC / JREP / JRID / JRSH. Beschreibung siehe Abschnitt "Job-Report-Parameter setzen" auf Seite **410**.

Überwachung von Enter-Aufträgen, die nicht von CFS gestartet wurden

Das Job-Report-System von CFS bietet die Möglichkeit, Jobs zu überwachen, die nicht von CFS durch das E-Kommando gestartet wurden. Als Voraussetzung für die Überwachung durch CFS müssen drei Bedingungen erfüllt sein:

1) Die zu überwachenden Enter-Aufträge müssen mit einer Monitorjobvariablen gestartet werden. Der Name der Monitorjobvariablen sollte folgendermaßen lauten: CFS.JR.*jrid*.*xxxx* . *jrid* ist die Job-Report-ID, unter der die Aufträge überwacht werden sollen. *xxxx* ist ein frei wählbarer Namenszusatz.

2) Vor dem Programmaufruf von CFS muß vom Benutzer eine Jobvariable mit dem Namen CFS.JR.*jrid*.JRINIT eingerichtet werden. *jrid* ist die Job-Report-ID, unter der die Enter-Aufträge überwacht werden sollen.

3) Nach dem erfolgreichen Start eines Enter-Auftrags muß in der startenden Instanz oder innerhalb des Enter-Auftrags die Jobvariable CFS.JR.*jrid*.JRINIT in Spalte 1 und 2 der Wert '$T' eingesetzt werden. Optional kann in Spalte 129 der Wert '#' und in Spalte 131 in der Länge 54 der Name der mit dem Enter-Auftrag verbundenen Monitorjobvariablen eingetragen werden.

Die Überwachung der Enter-Aufträge geschieht folgendermaßen:

Nachdem ein Enter-Auftrag gestartet wurde und die mit ihm verknüpfte Monitorjobvariable eingerichtet ist, wird die Jobvariable CFS.JR.*jrid*.JRINIT auf den Wert '$T' gesetzt. In dem Programm CFS, welches diese Job-Report-ID überwacht, wird dadurch ein Ereignis ausgelöst: Das Job-Report-System setzt intern ein JR-Kommando ab.

Damit werden ab diesem Zeitpunkt auch alle neu hinzugekommenen Jobvariablen CFS.JR.*jrid.xxxx* überwacht. Falls in der JRINIT-Jobvariablen der Name einer Monjv eingetragen wurde, so wird das interne JR-Kommando selektiv auf genau diese Monjv angewandt.

Überwachen benutzerdefinierter Ereignisse durch Jobvariablen

Das Job-Report-System von CFS bietet die Möglichkeit, beliebige, mit einer Jobvariablen verknüpfte Ereignisse zu überwachen und über die Jobvariable oder eine Datei Nachrichten an den Empfänger zu übermitteln.

Die von JRS ausgewerteten Jobvariablen müssen den folgenden Konventionen entsprechen:

Name der JV: CFS.JR.*jr-id.xxxx*

jr-id Job-Report Id des Benutzers, der die mit dieser JV verknüpften Ereignisse überwacht. Standardmäßig ist die JR-Id auf den Jobnamen des LOGON-Kommandos gesetzt. Ansonsten kann der Benutzer mit dem Kommando JRID auch eine beliebige andere JR-Id festlegen.

xxxxx vom Benutzer frei vergebener Namensteil.

Die verwendete Jobvariable muß nicht als Monitor-Jobvariable erklärt sein. Damit ist es dem Anwender möglich, über SETJV-Kommandos bzw. SETJV-Makroaufrufe die Position 1 der Jobvariablen selbst mit einem Inhalt z.B. '$T' (siehe unten) zu beschreiben und dadurch ein Ereignis für das Job-Report System auszulösen.

Inhalt der Jobvariablen

Spalte	Länge		
1	2	$T	normal Termination (aktiviert den Job-Report)
		$A	abnormal Termination (aktiviert den Job-Report)
		$R	running (für STAJ-Ausgabe)
		$S	Queue (für STAJ-Ausgabe)
			Die Bedeutung der einzelnen Werte wird weiter unten erklärt
3	126		nicht benutzt
129	2	#F	es folgt der 54 Byte lange Name einer Job-Report-File. Der Inhalt dieser Datei wird dem Job-Report Benutzer nach Beschreiben der Jobvariablen mit $T/$A zugestellt.
		#␣	es folgt eine max. 80 Byte lange Nachricht, die dem überwachenden Benutzer nach Beschreiben der Jobvariablen mit $T/$A in der Systemzeile eingeblendet wird.
131	54		Name der Job-Report File, falls im vorangehenden Feld '#F' eingetragen wurde.
185	26		Kommando des Job-Report Systems, das bei Ausgabe der Job-Report Maske ausgeführt wird. Es kann z.B. ein bestimmter Satz in der Sysout-Datei gesucht werden. Für weitere Informationen siehe Kommando JRIC.
131	80		Text, der dem Benutzer bei Eintritt des Ereignisses in die Systemzeile eingeblendet wird.

211	40		Job-Id. Dies ist ein freier Text, über den das Ereignis beim STAJ-Kommando identifiziert werden kann. Es kann hier auch eine bis zu 40 Stellen lange Nachricht eingetragen werden, die im STAJ über den momentanen Zustand des Enter-Auftrags Auskunft gibt.
251	4		frei
255	1	N	keine erneute Überwachung nach Quittierung der Job-Report Maske mit der K1-Taste oder Eingabe E (Ende).
256	1	C	Nach Quittierung der Job-Report Maske mit der K1-Taste oder Eingabe E (Ende) werden in der Jobvariablen die Felder Job-Id (Spalte 211, Länge 40) und Benutzertext bzw. Name der JR-File (Spalte 131, Länge 80) auf Blank gelöscht.

Setzen eines Ereignisses

Um überwacht zu werden, muß die Jobvariable bereits existieren, wenn der Benutzer CFS aufruft oder ein JR-Kommando absetzt. Soll dem überwachenden Benutzer über die Jobvariable ein Ereignis zugestellt werden, so sind folgende Schritte notwendig:

Mit aufeinanderfolgenden SETJV-Kommandos sind in der Jobvariablen die Felder an den Positionen ab 129 mit den gewünschten Inhalten zu beschreiben.

Das Ereignis wird dem Benutzer zugestellt (Job-Report Maske mit Anzeige der JR-File bzw. Einblenden der 80-stelligen Nachricht in die Systemzeile), sobald in die Position 1 der Jobvariablen der String '$T' bzw. '$A' geschrieben wird.

Mögliche Benutzeraktionen bei Benachrichtigung über Ereignis

Hier sind zwei Fälle zu unterscheiden:

a) Es wird die Job-Report Maske ausgegeben, da in der Jobvariablen der Name einer anzuzeigenden JR-File eingetragen wurde.

   Bei Quittierung der Job-Report Maske durch den überwachenden Benutzer mit der K1-Taste wird von JRS der String '$S' an die Position 1 der Jobvariablen geschrieben. Es erfolgt automatisch eine Überwachung dieser Jobvariablen (ONEVT). Die Jobvariable kann somit erneut zur Benachrichtigung verwendet werden. Nebeneffekt: Die Jobvariable ist belegt und kann nicht gelöscht werden. Falls keine neue Überwachung durch diese Jobvariable gewünscht wird, muß der letzte Job-Report mit dem Kommando EN beendet werden.

   Bei Eingabe des Kommandos N in der Job-Report Maske wird die Jobvariable von JRS unverändert gelassen. Der Vorgang "Benachrichtige Benutzer über Eintreffen eines Ereignisses" wird als noch nicht abgeschlossen angesehen. Beim nächsten JR-Kommando bzw. beim nächsten CFS-Aufruf wird dem Benutzer das Ereignis dann wieder vorgelegt.

b) Die auszugebende Nachricht wurde direkt in die Jobvariable geschrieben (Position 131 - 210) und dem überwachenden Benutzer in der Systemzeile eingeblendet. Ansonsten gelten die Ausführungen unter Punkt a).

Kommandos zur Steuerung des Job-Report Systems

Mit den folgenden Parametern werden globale Job-Report Parameter gesetzt bzw. verändert. Die Kommandos können in der Kommandozeile der Dateienliste, Job-Report Parameter-Maske (E-Kommando), sowie in der Job-Report Maske (SYSOUT-Listing eines beendeten Enter-Prozeßes) eingetragen werden.

Job-Report aktivieren / deaktivieren

JR [$*userid* | *jr-id*] [$*user-id* . *jr-id*] [*RES]

Es wird die Überwachung für alle mit unter der angegebenen *jr-id* gestarteten Enter-Aufträge eingeschaltet. Die bereits beendeten Aufträge werden sofort angezeigt. Es werden nur die Aufträge überwacht, die zum Zeitpunkt des JR-Kommandos bereits gestartet waren. Wenn keine *jr-id* angegeben wird, so wird die zur Zeit gültige *jr-id* oder eine durch das Kommando JRID gewählte *jr-id* aktiviert.

Das Kommando kann beliebig oft mit verschiedenen *jr-id* wiederholt werden. In diesem Fall bleiben alle zuvor angegebenen *jr-id*'s aktiv. *jr-id* kann auch in der JRP-Maske (Kommando JRP) angegeben werden.

JR-id's von fremden Benutzerkennungen ($*user-id*) sind nur dann sinnvoll, wenn unter der fremden Benutzerkennung vor dem Start der Jobs das Kommando JRSH bzw. bei JRP die Option SHARE=YES gewählt wurde.

*RES Restore. Nach einem NJR-Kommando können mit JR *RES alle zuvor aktiven JR-ID's wieder hergestellt werden.

Wurde ein STAJ-Kommando ohne *jr-id* eingegeben, so werden die überwachten Jobs aller aktivierten Job-Report-Id's ausgegeben.

Standard für JRID: Jobname bzw. Abrechnungs-Nr., falls kein Jobname vergeben wurde.

NJR Die Job-Report Überwachung wird global, d.h. für alle *jr-id*'s ausgeschaltet.

Enter-Parameter vorbelegen

JREP *enter-params* Job-Report Enter-Parameter.

Für alle mit dem Job-Report System gestarteten Aufträge werden im Enter-Kommando die angegebenen Parameter verwendet.

enter-params Es können alle im BS2000 erlaubten Enter-Parameter angegeben werden. Diese Parameter gelten dann für alle nachfolgenden E-Kommandos. Die durch das JREP-Kommando vorbelegten Enter-Parameter können im E-Kommando und/oder in der Parameter-Maske für DO-Prozeduren noch durch weitere Enter-Parameter ergänzt werden. Die Enter-Parameter können auch in der JRP-Maske (Kommando JRP, Seite 414) eingegeben werden.

Job-Report-File festlegen

JRF [*datei* | *LIB(*bibl*,*element*[(*vers*)],*typ*)] [,E] [, SYSOUT | SYSLST | JRFILE]

Das JRF-Kommando bewirkt, daß für alle nachfolgend zu startenden Jobs die angegebene Datei bei Beendigung des Jobs in der Job-Report Maske angezeigt wird. Die Parameter dieses Kommandos können im selben Format auch im E-Kommando angegeben werden.

Die Angaben im JRF-Kommando, im JRF-Parameter des E-Kommandos bzw. im /REMARK-Kommando in der zu startenden Prozedur haben folgende Wirkung:

datei
Die angegebene Datei soll bei der Beendigung aller nachfolgend mit dem E-Kommando gestarteten Enter-Aufträge angezeigt werden. Dies kann z.B. sinnvoll sein, wenn alle Enter-Aufträge ihr Protokoll in die gleiche, evtl. mit EXTEND eröffnete, Datei schreiben.
Zur Bildung des Dateinamens können die folgenden Variablen verwendet werden:

&TSN	TSN des CFS-Tasks, der den Enter-Auftrag startet.
&DATE	Startdatum des Enter-Auftrags im Format yymmdd.
&TIME	Startzeitpunkt des Enter-Auftrags im Format hhmmss.
&LOGNAM	Jobname des CFS-Tasks, der den Enter-Auftrag startet.

*LIB(*bibl*,*element*[(*vers*)] ,*typ*
Das angegebene Bibliothekselement wird angezeigt (ab OSD V1).

E
Die SYSOUT-/SYSLST-Datei wird im Extend-Modus eröffnet.

SYSOUT | SYSLST
Die SYSOUT-/SYSLST-Datei aus dem ersten in der Prozedurdatei enthaltenen SYSFILE SYSOUT/ASSIGN SYSOUT/SYSFILE SYSLST/ASSIGN SYSLST-Kommando soll in der Job-Report-Maske angezeigt werden. Die Zuweisung von SYSLST bzw. SYSOUT muß in den ersten 20 Sätzen nach dem LOGON-Kommando vorkommen.

Bei einer DO-Prozedur, die im Enter-Modus gestartet wird und die kein entsprechendes SYSFILE-Kommando enthält, generiert CFS ein SYSFILE-Kommando in der übergeordneten Enter-Datei in folgender Form:

/SYSFILE SYSOUT=CFS.SYSOUT.tsn.time bzw.
/SYSFILE SYSLST=CFS.SYSLST.tsn.time.
Beim Starten einer DO-Prozedur im Enter und JRF=SYSLST wird in der Enter-Datei neben dem Kommando /SYSFILE SYSLST auch /OPTION MSG=FH eingefügt. Damit wird erreicht, daß auch die SYSOUT-Ausgaben in der SYSLST-Datei enthalten sind.

JRFILE
Die Datei aus dem ersten Kommando /REMARK JRF=datei in der Prozedurdatei soll in der Job-Report Maske angezeigt werden.

Standard für JRF: SYSOUT.

Beispiele:

```
JRF CFS.SYSOUT.&LOGNAM.&DATE.&TIME
```

```
E CFS.DO.TEST,PAR1=XY/TIME=10,JRF=CFS.FILE.XYZ
```

```
E CFS.DO.TEST,PAR1=XY/TIME=10,JRF=*LIB(CFS.LIB.PRINT,
FILE.XYZ,P),SYSLST
```

Hinweise:

Falls über das E-Kommando eine DO-Prozedur im Enter gestartet wird und der Parameter JRF=*datei* angegeben ist, wird in der von CFS erzeugten Enter-Datei keine

Anweisung /ASSIGN-SYSOUT TO-F=*datei* eingefügt. Ausnahme: Bei der Definition des Dateinamens wurden mindestens eine der Variablen &TSN, &DATE, &TIME, &LOGNAM verwendet.

Die Parameter des JRF-Kommandos können auch in einem REMARK-Kommando in der DO- oder Enter-Prozedurdatei angegeben werden.
Beispiel: /REMARK JRF=CFS.FILE.XYZ.
Das REMARK-Kommando muß in den ersten 20 Sätzen der Enter-/Proc-Datei angegeben werden. Falls ein REMARK JRF-Kommando in der DO-/Enter-Prozedur vorkommt, so gelten die JRF-Angaben nur für den aktuellen Enter-Auftrag.

Die verschiedenen Quellen für die Angaben zur JR-File werden mit folgender Priorität ausgewertet (1 ist hierbei die höchste Priorität):

1) Explizite Angabe der JR-File in der Parameter-Maske.

2) Explizite Angabe der JR-File im E-Kommando.

3) JR-File aus dem /REMARK-Kommando bzw. dem SYSFILE SYSOUT/ SYSLST-Kommando in der Prozedurdatei.

4) Datei aus dem Job-Report-Kommando JRF bzw. der JRP-Maske.

5) Der von JRS im generierten SYSFILE-Kommando eingetragene Standardname CFS.SYSOUT.tsn.time / CFS.SYSLST.tsn.time.

Der Name der zur Anzeige im Job-Report ausgewählten Datei wird in der Parameter Maske (siehe Seite **403**) angezeigt und kann dort noch verändert werden.

Job-Report bei Beendigung von File-Transfer Aufträgen aktivieren

JRFT

Das JRFT-Kommando bewirkt, daß alle nachfolgend über CFS gestarteten File-Transfer Aufträge, bei denen in der Maske der Parameter PROTOCOL-LISTING: YES gesetzt wurde, mit einer Monitor-Jobvariablen verknüpft werden. Dies hat zur Folge, daß bei Beendigung der FT-Aufträge das Protokoll über den Job-Report von CFS angezeigt wird.

Hinweis:

Nach Einschalten der JRFT-Option ist der Benutzer i.a. nicht mehr an den asynchronen Ausführungsmeldungen des openFT ("FILE TRANSFERRED. TRANSFER-ID xxxx") interessiert. Die Ausgabe dieser Messages kann unterdrückt werden durch das BS2000-Kommando /BCNTRL MES=N. Durch das angegebene Kommando werden alle Messages, insbesondere auch vom Operator mit dem Kommando /MES *tsn*,... gesendete Messages an den eigenen Benutzerprozeß unterdrückt. Die Ausgabe von Broadcasts wird damit nicht beeinflußt.

NJRFT

File-Transfer Aufträge werden nicht mit einer Monitor-Jobvariablen und somit auch nicht mit dem Job-Report von CFS verknüpft. Das gleiche gilt auch, falls in der File-Transfer Maske von CFS der Parameter PROTOCOL-LISTING auf dem Standardwert NO belassen wurde. In BS2000-Systemen mit einer FT-Version kleiner als 3 ist die Einstellung NJRFT ebenfalls Standard. Der Einsatz der FT-Version 2 muß CFS über den Modul CFSMAIN bekannt gemacht werden. Siehe hierzu CFS-Handbuch für Systemverwalter, Abschnitt "Installation von CFS, ausgelieferte Dateien".

Job-Report Initial Command

JRIC *cmd*

Job-Report Initial Command.
Bei Ausgabe einer Job-Report Maske nach Beendigung eines Enter-Auftrags wird als erstes das angegebene Kommando in der Job-Report-Maske ausgeführt. JRIC eignet sich dafür, die Anzeige der JR-File gleich zu Anfang ab einer bestimmten Stelle beginnen zu lassen bzw. nach einer Zeile mit einem bestimmten String zu suchen. Damit kann ein umständliches Blättern oder Suchen bis zur gewünschten Information abgekürzt werden.

cmd

Jedes im Kommandofeld der Job-Report Maske angebbare Kommando.
Für eine Zusammenstellung aller möglichen Kommandos siehe Seite 405.

Beispiele:

```
JRIC ++
```

```
JRIC S,'FLAGS IN '
```

Hinweis:

Die Parameter des JRIC-Kommandos können auch in einer REMARK-Anweisung der Enter-Datei oder DO-Prozedur angegeben werden. Beispiel: /REMARK JRIC=*cmd*
Das REMARK-Kommando muß in den ersten 20 Sätzen nach dem LOGON, PROC oder START-PROC-Kommando in der DO-/Enter-Prozedur angegeben werden. Falls ein REMARK JRIC-Kommando in der DO-/Enter-Prozedur vorkommt, so gelten die JRIC-Angaben nur für den aktuellen Enter-Auftrag.

JR-Identifikation festlegen

JRID *jr-id*

interne JR-Identifikationskennung (JRID) festlegen.

jr-id

Vom Benutzer frei wählbare, max. 8 Stellen lange interne Identifikation, mit der alle nachfolgend zu startenden Enter-Aufträge verknüpft werden und unter der diese auch überwacht werden können (STAJ *jr-id*).

Falls CFS vor Abschluß eines mit E ... gestarteten Enters beendet und zu einem späteren Zeitpunkt wieder aufgerufen wird, so gilt zunächst die Standard Job-Report-ID, d.h. der Benutzer wird standardmäßig über den Abschluß aller unter der JR-ID *jobname* bzw. *acct-nr* gestarteten Enter-Aufträge informiert.

Dies kann zu ungewollten Effekten führen, falls mehrere Benutzer unter der gleichen Benutzerkennung und Abrechnungsnummer bzw. mit dem gleichen Jobnamen arbeiten. Um diesem Fall vorzubeugen, kann jeder Benutzer nach dem Aufruf von CFS mit dem JRID-Kommando eine benutzerindividuelle JRID festlegen.

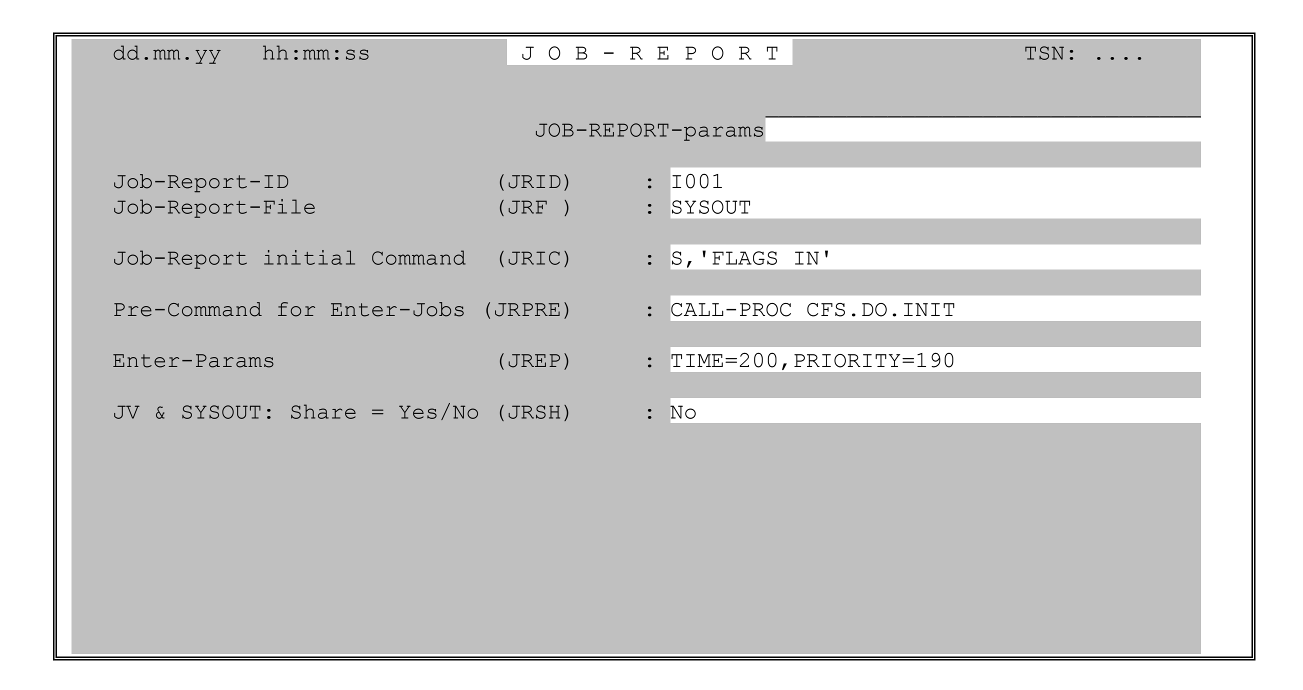
Das JRID-Kommando bewirkt, daß nachfolgend gestartete Enter-Aufträge unter der angegebenen Job-Report-ID überwacht werden. Das JRID-Kommando bewirkt nicht, daß bereits früher mit dieser JRID gestartete Enter-Aufträge bei Beendigung automatisch angezeigt werden. Hierzu ist das Kommando JR zu verwenden.

Standard für JRID: Jobname des Dialogprozeßes, unter dem CFS geladen wurde. Falls im LOGON-Kommando kein Jobname angegeben wurde, so wird als *jr-id* die Abrechnungsnummer der Benutzerkennung angenommen.

Maske der Job-Report Parameter

JRP Aufgrund dieses Kommandos wird eine Maske ausgegeben, in der die aktuellen Werte für die Job-Report-Parameter JRID, JRF, JREP, JRSH und JRIC angezeigt werden. Die in der JRP-Maske angezeigten Parameterwerte können vom Benutzer modifiziert werden.

Die JRP-Maske hat folgendes Aussehen:



```
dd.mm.yy   hh:mm:ss          J O B - R E P O R T              TSN: ....

                                JOB-REPORT-params

Job-Report-ID               (JRID)     : I001
Job-Report-File             (JRF )     : SYSOUT

Job-Report initial Command  (JRIC)     : S,'FLAGS IN'

Pre-Command for Enter-Jobs  (JRPRE)    : CALL-PROC CFS.DO.INIT

Enter-Params                (JREP)     : TIME=200,PRIORITY=190

JV & SYSOUT: Share = Yes/No (JRSH)     : No
```

Einzufügendes Kommando vor CALL auf DO-Prozedur

JRPRE *cmd* Falls über das Job-Report System eine DO-Prozedur im Enter ausgeführt wird, kann mit JRPRE ein BS2000-Kommando definiert werden, das im Enter-Job vor dem CALL-PROC-Kommando ausgeführt wird.

Standard: Es wird kein zusätzliches Kommando vor dem CALL-PROC auf die DO-Prozedur ausgeführt.

Mehrfachbenutzbarkeit der Job-Report-Files

JRSH Die vom JRS erzeugten Jobvariablen und SYSOUT/SYSLST-Dateien werden mit dem Attribut SHARE=YES erzeugt. Dadurch ist es möglich, auch von einer fremden Benutzerkennung die Job-Überwachung mit dem Kommando JR $*user-id* zu initialisieren oder den Status der von JRS überwachten Jobs abzufragen.

NJRSH Die von JRS erzeugten Job-Variablen und SYSOUT/SYSLST-Dateien werden mit dem Attribut SHARE=NO erzeugt. Im JR- und STAJ-Kommando können in diesem Fall keine JRID's von fremden Benutzerkennungen angegeben werden.

Standard: NJRSH. Der Systemverwalter kann die Standardeinstellung auf JRSH ändern (siehe CFSMAIN).

12. File-Transfer mit openFT/RDAC

CFS unterstützt die File-Transfer Produkte openFT und RDAC. Es können damit auf einfache Weise Dateien und PLAM-Bibliothekselemente (nur mit openFT) zwischen verschiedenen Host-Rechnern bzw. zwischen Host-Rechner und PC-Systemen (MS-DOS, SINIX) ausgetauscht werden. Mit RDAC können auch Dateien auf Stationsdruckern ausgegeben werden. Das Kommando RDAC ist auch dann anwendbar, wenn anstelle von RDAC das Produkt FTOS eingesetzt wird.

Der File-Transfer wird angeboten als Variable Action ONXFT ... / ONXRDAC ..., als Kommando FT/ RDAC und als Action-Code FT/RD. Bei der Definition der Variablen Action/des Kommandos FT/RDAC können die zum File-Transfer notwendigen Angaben wie z.B. Name des Ziel-Rechners, Benutzerkennung, Abrechnungsnummer und Logon-Passwort als Stellungsparameter angegeben werden. Wurden keine oder nicht alle benötigten Parameter angegeben, so wird vom FT-Modul automatisch eine sog. FT-Maske ausgegeben. In dieser können die fehlenden FT-Parameter ergänzt werden.

Die FT-Parameter werden gespeichert und stehen ab diesem Zeitpunkt für alle weiteren FT-Anforderungen als Standardwerte zur Verfügung.

Bei der ersten Angabe der Action-Codes FT/RD wird die FT-Maske ausgegeben, sofern die Standardwerte nicht bereits durch ein früheres Kommando FT/RDAC bzw. durch eine Variable Action ONXFT/ ONXRDAC gesetzt wurden. Für alle folgenden FT-/RD-Action-Codes werden die in der FT-Maske eingetragenen Werte verwendet. Mit den Action-Codes FTM/RDM kann die FT-Maske jederzeit wieder angefordert werden. Mit dem Action-Code FTR ist ein Umbenennen der Datei bzw. des Bibliothekselements im Zielsystem möglich.

File Transfer mit openFT

```
dd.mm.yy   hh:mm:ss       HOST: ........       TSN: ....  USER-ID: ........


                    Parameters for File-Transfer (FT)


Filename to be transferred :
Filename in Target-System  : *SAME
Password for remote File   :

Partner-Name                         :
Remote User-ID / FTAC:transfer-adm.  : AAAAAA1
Remote Account-#                     : I0000
Remote Password                      :
Remote Success-Procedure             :
Remote Failure-Procedure             :
Remote-Parameter (*BS2000/*MSP/*ANY) : *BS2000
Transfer-Direction         (FROM/TO) : TO
Protocol-Listing            (YES/NO) : NO
Start Time                   (hh:mm) :
modify Extended Params for FT  (y/n) : N
```

Für eine ausführliche Beschreibung der einzelnen Felder der FT-Maske und der entsprechenden Eingabemöglichkeiten wird auf das Kommando FT Seite 241 verwiesen.

Wird die FT-Maske aufgrund der Variablen Action ONXFT bzw. aufgrund des Action-Codes FT ausgegeben, so werden die Felder 'Filename to be transferred', 'Filename in Target-System' und 'Password for remote File' nicht angezeigt. Ausnahme: Es sind PLAM-Bibliothekselemente zu übertragen. In diesem Fall sind die Felder 'Libname in Target System' und 'Password for Remote File' am Bildschirm sichtbar. Als Vorbelegung ist hier der Name der Ursprungsbibliothek eingetragen. Falls diese Standardbelegung nicht verändert wird, so werden die Bibliothekselemente im Zielsystem in einer PLAM-Bibliothek gleichen Namens abgelegt.

Die Felder 'Password for Remote File' und 'Remote User-ID Password' sind in der FT-Maske aus Gründen des Datenschutzes dunkel gesteuert.

Werden in der FT-Maske Werte eingetragen, so stehen diese fortan als neue Standardwerte zur Verfügung.

In Batch-Anwendungen von CFS, in denen keine FT-Maske ausgefüllt werden kann, müssen die FT-Parameter als Stellungsparameter im Kommando FT bzw. bei der Definition der Variablen Action angegeben werden. Die Reihenfolge der Stellungsparameter entspricht der Reihenfolge der Maskenfelder.

Beispiel:

```
ONXFT HOST5,USER2,I2222,,
```

Der File-Transfer erfolgt zum Rechner HOST5 in die Kennung USER2 mit der Abrechnungsnummer I2222. Diese Kennung hat kein Logon-Passwort. Der entsprechende Parameter wurde daher durch ',,' auf seinen Standardwert gesetzt.

Die explizite Angabe der Parameter jenseits von *passw2* kann auch in Batch-Läufen von CFS entfallen. Es werden dann die Standardwerte verwendet.

Job-Report bei Beendigung von File-Transfer Aufträgen aktivieren

JRFT: Das JRFT-Kommando bewirkt, daß alle nachfolgend über CFS gestarteten File-Transfer Aufträge, bei denen in der Maske der Parameter Protocol-Listing : YES gesetzt wurde, mit einer Monitor-Jobvariablen verknüpft werden. Dies hat zur Folge, daß bei Beendigung der FT-Aufträge das Protokoll über den Job-Report von CFS angezeigt wird.

Hinweise:

Die Beendigung von FT-Aufträgen kann mit dem Job-Report System von CFS überwacht werden. Für weitere Informationen zu diesem Thema wird auf die Kommandos JRFT/ NJRFT auf Seite 412 verwiesen.

Zur Selektion aller mit openFT übertragenen Dateien wird auch auf die NSTAT-Option im Feld FILENAME-SELECT der Selektionsmaske hingewiesen.

Beim Versenden von Dateien (TRANSFER-DIRECTION : TO) werden die übertragenen Dateien in der in REMOTE USER-ID angegebenen Kennung im Zielrechner eingerichtet. Ausnahme: Der Benutzer hat in der FT-Maske bei REMOTE USER-ID : TSOS mit der entsprechenden Abrechnungsnummer und Passwort angegeben. In diesem Fall kann der Benutzer die Dateien in jede beliebige Kennung im Zielrechner übertragen durch Voranstellen einer Benutzerkennung vor den Namen der Datei im Remote-System.

Beim Abholen von Dateien (TRANSFER-DIRECTION : FROM) werden die übertragenen Dateien unter der eigenen Kennung eingerichtet. Ausnahme: Der Benutzer arbeitet unter der Kennung TSOS. In

diesem Fall können die Dateien in jeder beliebigen Kennung im eigenen Host eingerichtet werden durch Voranstellen einer Benutzerkennung vor den Namen der Datei im lokalen (eigenen) System.

File-Transfer von Jobvariablen

Mit openFT oder RDAC können Jobvariablen nicht auf direkte Weise auf andere Host-Rechner übertragen werden. Falls CFS sowohl auf dem Senderechner, wie auch auf dem Empfangsrechner installiert ist, so ist es damit möglich, Jobvariablen von einem Hostrechner auf einen anderen zu übernehmen.

Folgende Vorgehensweise ist dazu notwendig:

a) Auf dem Senderechner sind die gewünschten Jobvariablen mit der Variablen Action ONXLISTCAT in einer Datei (sog. Listcat-Datei) sicherzustellen. LISTCAT sichert sowohl den Katalogeintrag, als auch die Daten der Jobvariablen.

b) Die so erzeugte Listcat-Datei ist sodann per File-Transfer (openFT/ RDAC) auf den Zielrechner zu transferieren, auf dem die Jobvariablen später benötigt werden.

c) Auf dem Zielrechner ist die übertragene Listcat-Datei durch das Kommando RECAT weiter zu verarbeiten: RECAT *datei*,CJ.
Mit diesem Kommando werden die Jobvariablen auf dem Zielrechner eingerichtet und mit den ursprünglichen Daten versorgt.

Datenaustausch mit SINIX-Systemen über openFT

Mit FT können Dateien zwischen BS2000 und SINIX-Rechnern ausgetauscht werden.

Variable Action ONXFT:
In der Parameter-Maske für die Variable Action ONXFT sind u.a. folgende Eingaben möglich:

```
Partner-Name              : mx500          BCAM-Name des Sinix-Rechners
Remote User-ID            : 'gast'
Remote Account-#          :
Remote Password           : 'siemens'
Remote-Parameter          : *any
Transfer-Direction        : TO
```

TRANSFER-DIRECTION : TO Die zu übertragenden Dateien werden im SINIX-System unter dem User /usr/gast eingerichtet.

TRANSFER-DIRECTION : FROM Es werden die angegebenen Sinix-Dateien von der Kennung /usr/ gast geholt.

Der Benutzer kann bei Remote User-Id auch den vollständigen Pfad angeben, z.B. /usr2/lkz/gast.

TRANSFER-DIRECTION : TO Es werden die Dateinamen im Zielsystem gebildet, indem die ersten 12 Zeichen der Namen der zu übertragenden Dateien an den unter Remote User-ID angegebenen Pfad angehängt werden. Die BS2000-Dateinamen werden in Kleinbuchstaben umgewandelt (Standard).

Wird in der Maske im Feld Remote-Parameter : *any in Kleinbuchstaben angegeben, so werden die BS2000-Dateinamen dem Sinix-System in Kleinbuchstaben übergeben.

Wird in der Maske im Feld Remote-Parameter : *ANY in Großbuchstaben angegeben, so werden die BS2000-Dateinamen dem Sinix-System in Großbuchstaben übergeben.

Kommando FT:

```
Filename to be transferred : cfs.d.tabelle
Filename in Target-System  : '/usr/lkz/gast/tabelle'
Partner-Name               : mx500          BCAM-Name des Sinix-Rechners
Remote User-ID             : 'lkz'
Remote Account-#           :
Remote Password            : 'siemens'
Remote-Parameter           : *any
Transfer-Direction         : TO
```

Im FT-Kommando kann der Name der einzurichtenden bzw. abzuholenden SINIX-Datei als vollständiger Pfadname angegeben werden (z.B. '/usr/lkz/gast/tabelle'). Der Dateiname wird ohne Veränderung an das SINIX-System weitergereicht. Die zu übertragende Datei kann somit unter jedem beliebigen User eingerichtet werden bzw. von jedem (beliebig tief geschachtelten) User abgeholt werden.

Hier ist ebenfalls die Schreibweise von *ANY im Feld REMOTE PARAMETER zu beachten:

*any   Kleinbuchstaben im Dateinamen werden als solche an das SINIX-System weitergegeben.

*ANY   Kleinbuchstaben im Dateinamen werden dem SINIX als Großbuchstaben übergeben.

Im Feld REMOTE USER-ID ist ein Benutzer anzugeben, der die zu übertragende Datei im Sinix-System anlegen darf (TRANSFER-DIRECTION : TO) bzw. ein User, der die Datei lesen darf (TRANSFER-DIRECTION : FROM).

Die Angaben für Remote User-ID und Remote User-ID Password sind in Hochkommas einzuschließen und werden - unabhängig von der Schreibweise von *any/*ANY - an das SINIX-System so wie in der Maske eingetragen (d.h. im allgemeinen in Kleinbuchstaben) übergeben.

File Transfer mit RDAC

```
   dd.mm.yy   hh:mm:ss       HOST: ........        TSN: ....  USER-ID: ........


                         Transfer-Parameters for RDAC


   Filename to be transferred :
   Filename in Target-System  : *SAME
   Password for remote File   :

   Partner-Name                        :
   Remote User-ID                      : AAAAAA1
   Remote Account-#                    : I0000
   Remote Password                     :
   Remote Success-Procedure            :


   Transfer-Direction         (FROM/TO) : TO
   Print ?  (NO/YES/form <,pr-params> ) : NO
   Start Time                   (hh:mm) :
```

Für eine ausführliche Beschreibung der einzelnen Felder der FT-Maske und der entsprechenden Eingabemöglichkeiten wird auf das Kommando RDAC Seite 265 verwiesen.

Wird die RDAC-Maske aufgrund einer var. Action-Anforderung ONXRDAC, bzw. aufgrund des Action-Codes RD ausgegeben, so werden die Felder 'Filename to be transferred', 'Filename in Target-System' und 'Password for remote File' nicht angezeigt.

Die Felder 'Password for Remote File' und 'Remote User-ID Password' sind in der FT-Maske aus Gründen des Datenschutzes dunkel gesteuert.

Werden in der FT-Maske Werte eingetragen, so stehen diese fortan als neue Standardwerte zur Verfügung.

In Batch-Anwendungen von CFS, in denen keine FT-Maske ausgefüllt werden kann, müssen die FT-Parameter als Stellungsparameter im Kommando RDAC bzw. bei der Definition der Variablen Action angegeben werden. Die Reihenfolge der Stellungsparameter entspricht der Reihenfolge der Maskenfelder.

Hinweise:

Das Kommando RDAC ist auch dann anwendbar, wenn anstelle von RDAC das Produkt FTOS eingesetzt wird. Das gleiche gilt auch für die Variable Action ONXRDAC sowie Action-Code RD.

Der Parameter Print ? : YES bewirkt, daß die angegebene Datei auf dem im Maskenfeld 'Partner-Name' spezifizierten Remote-Drucker ausgegeben wird.
Die Felder der FT-Maske: 'Remote User-ID', 'Remote User-ID Account-#' und 'Remote User-ID Password' brauchen bei Ausgaben an einen Remote-Drucker nicht ausgefüllt zu werden.

Bei bestimmten von RDAC bedienten Druckern, insbesondere bei Tischlaserdruckern, kann eine Datei mit unterschiedlichen Layouts gedruckt werden. Die verschiedenen Layouts werden beim Ausdruck über den Form-Parameter ausgewählt.

Bei Angabe von YES im Maskenfeld 'Print ?' wird der Standard Form-Parameter 'STD' verwendet. Die Angabe eines Wertes ungleich YES/NO im Maskenfeld 'Print ?' wird als Form-Parameter interpretiert. In diesem Fall wird automatisch angenommen, daß es sich bei der Ausgabe um einen Druckvorgang handelt (Print = YES).

Bei Angabe von YES | *form* im Feld "Print ?" können noch zusätzliche Print-Parameter angegeben werden. Erlaubt sind die folgenden Print-Parameter: SPACE=., COPIES=.., HEADER=.., LINES=.. Durch den Action-Code RDS werden Dateien mit SPACE=E am Remote-Drucker ausgegeben.

Datenaustausch mit PC-Systemen (MS-DOS) über RDAC

Mit RDAC/FTOS können Dateien zwischen BS2000 und MS-DOS PC's ausgetauscht werden.

Kommando RDAC:

```
Filename to be transferred : cfs.d.tabelle
Filename in Target-System  : C:~subdir~tabelle
Partner-Name               : pcd2                    BCAM-Name des PC
Remote User-ID             :
Remote Account-#           :
Remote Password            :
Transfer-Direction         : TO
Print ?                    : NO
```

Im RDAC-Kommando kann der Name der einzurichtenden bzw. abzuholenden MS-DOS Datei als vollständiger Pfadname incl. Laufwerk angegeben werden. Der Name der MS-DOS Datei ist nicht in Hochkommas einzuschließen. Die zu übertragende Datei kann somit in jedem beliebigen Subdirectory eingerichtet werden bzw. von jedem (beliebig tief geschachtelten) Subdirectory abgeholt werden.

Die Angaben für Remote User-ID, Remote User-ID Account-# und Remote User-ID Password sind bei der Übertragung von Dateien von/nach MS-DOS Rechnern ohne Belang und daher auf Blanks zu löschen.

Versenden von Dateien an mehrere Rechner

1) Auswahl der in Frage kommenden Dateien durch entspr. Eingaben in der Selektionsmaske.

2) `ONXFT host1 [,usid,acct-#,passw]`
Definition der Variablen Action für den ersten Host-Rechner.

3) Ankreuzen der zu versendenden Dateien mit dem Action-Code X.

4) Nach Ausführung der Variablen Action: Beantwortung der Terminierungsabfrage mit U (Updated List). Die eingetragenen X-Action Codes bleiben in der Dateienliste erhalten.

5) `ONXFT host2 [,usid,acct-#,passw]`
Definition der Variablen Action für den zweiten Zielrechner. Achtung: Das ONX-Kommando darf nicht mit einer Endemarke abgeschlossen werden, da sonst die in der Dateienliste eingetragenen X-Action Codes gelöscht werden.

6) Ausführung der Variablen Actions mit dem Kommando A starten.

Die Punkte 4 bis 6 können beliebig oft wiederholt werden.

Falls wenige Dateien in einer Dateienliste an verschiedene Rechner zu übertragen sind, so empfiehlt sich folgende Vorgehensweise:

1) Die erste Datei mit Action-Code FTM markieren (explizite Anforderung der FT-Parameter Maske) und alle übrigen mit FT markieren.

2) In der aufgrund von FTM ausgegebenen Maske werden die Parameterangaben für den ersten Zielrechner, Benutzerkennung usw. eingetragen.

3) Kommando A (Ausführen Actions), falls die Dateienliste mehr als eine Bildschirmseite umfaßt.

4) Terminierungsabfrage mit U oder ENTER beantworten. Anschließend obige Prozedur für den zweiten Zielrechner wiederholen.

13. Datei-Informationssystem

Mit dem Dateiinformationssystem von CFS kann zu jeder Datei, Jobvariablen und zu jedem Element einer Bibliothek ein beliebiger Text gespeichert werden. Dadurch ist es z.B. möglich, über die Verwendung jeder einzelnen Datei Buch zu führen. Diese einmal erfaßten Dokumentationen können am Bildschirm jederzeit abgefragt und aktualisiert werden. Die Dokumentationen zu den Datenobjekten werden in von CFS angelegten und verwalteten Datenkatalogen gespeichert.

Zur Aufnahme der Dokumentationstexte für Dateien einer Benutzerkennung, sowie für die Aufnahme der Dokumentationen der Elemente einer Bibliothek, wird jeweils ein eigener Datenkatalog angelegt. Der Datenkatalog wird bei der ersten Eingabe einer Dokumentation angelegt (Action-Code IM).

Falls der Benutzer kein explizites File-Kommando angegeben hat, so wird der Name des CFS-Datenkatalogs folgendermaßen gebildet:

CFS.INF	Datenkatalog für Dateien der eigenen Benutzerkennung.
CFS.INF.JV	Datenkatalog für Jobvariable der eigenen Benutzerkennung.
bibl.INF	Datenkatalog für Elemente einer Bibliothek *bibl*.

Für Dateien und Jobvariablen kann unter jeder Benutzerkennung je ein Datenkatalog existieren. Ebenso kann für jede Bibliothek ein Datenkatalog existieren.

Mit einem FILE-Kommando (LINK=INFO) kann der Name des Datenkatalogs vom Benutzer auch fest vorgegeben werden.

Die Eingabe der Dokumentationen erfolgt normalerweise formatfrei, d.h. der Benutzer bekommt einen leeren Bildschirm angeboten, in den er seine Dokumentationstexte einträgt.

Die Dokumentationen können auch formatgesteuert über Bildschirmmasken eingegeben werden. Diese INF-Masken müssen in einem zentralen INF-Formatkatalog (ISAM-Datei) vorliegen und können über ein- bis dreistellige Formatnamen angesprochen werden. Der INF-Formatkatalog hat per Voreinstellung den Namen $TSOS.CFS.INF.FORMAT. Per File-Kommando kann der Benutzer jedoch seinen eigenen Formatkatalog zuweisen: /FILE *datei*,LINK=IFORMAT.

Die Neuerfassung von Dokumentationstexten über ein Format geschieht durch Anhängen des ein- bis dreistelligen Formatnamens *f* an den Action-Code IM/IA (Information Modify/ Information Add). Der Systemverwalter kann die Voreinstellung von CFS dahingehend ändern, daß die Neuerfassung von Dokumentationen mit einem fest vorgegebenen Standardformat (*x*) verknüpft ist (implizites IM*x*).

Action-Codes zum Erstellen/Abfragen des CFS-Datenkatalogs

I Inform. Anzeigen der zu dem Datenobjekt (Datei/Jobvariable/Bibliothekselement) eingegebenen Dokumentation.

IM [*f*] Information Modify. Neuerfassung/Ändern eines Dokumentationstextes.

f Nur bei Neuerfassung eines Dokumentationstextes: 1- bis 3-stelliger mnemotechnischer Name eines Formats für die Erfassung des Dokumentationstextes. Die Maskenformate für die Erfassung von Dokumentationstexten und die Mnemo-Codes, über die diese Masken angesprochen werden, sind vom Systemverwalter in einem zentralen Katalog CFS.INF.FORMAT einzutragen.

IA [*f*] Information Add. Anfügen einer zusätzlichen Seite von Dokumentationstext.

f 1- bis 3-stelliger Name eines Formats für die Erfassung der Dokumentation.

Falls für das betreffende Datenobjekt noch keine Dokumentation gespeichert ist, so wirkt der Action-Code IA wie IM (Neuerfassung).

II [*f*] Information Insert. Einfügen einer zusätzlichen Seite von Dokumentationstext. Die neue Seite an Informationstext wird hinter der letzten, mit dem gleichen Format erfaßten Informationsseite eingefügt.

f 1- bis 3-stelliger Name eines Formats für die Erfassung der Dokumentation.

Hinweis: Falls für das betreffende Datenobjekt noch keine Dokumentation gespeichert ist, so wirkt der Action-Code II wie IM (Neuerfassung). Falls das im II Action-Code angegebene Format im bestehenden Informationstext nicht vorkommt, so wird die neue Seite an Informationstext am Ende der Beschreibung eingefügt (gleiche Wirkung wie IA: Information Add).

IE Information Erase. Löschen der zu einem Datenobjekt gespeicherten Dokumentationen.

IEQ Information Erase with Query. Die einzelnen Seiten des Informationstextes werden wie beim Action-Code I (Inform) angezeigt. Die angezeigte Informationsseite wird durch Drücken der ENTER-Taste gelöscht. Die Löschung der angezeigten Informationsseite wird durch Betätigung der K3-Taste unterbunden. In beiden Fällen (ENTER/K3) wird dem benutzer die nächste Informationsseite angeboten. Durch Drücken der K1-Taste kann der Vorgang abgebrochen werden. Die zuletzt angezeigte Informationsseite wird in diesem Fall nicht gelöscht.

IC Information Copy. Kopieren des gesamten Informationstextes. Es wird die Eingabe des Namens eines bereits dokumentierten Datenobjekts verlangt. Die von diesem Datenobjekt kopierte Dokumentation kann anschließend modifiziert werden (implizites IM).

Hinweise:

Falls der Benutzer im Action-Code IM/IA den Namen eines nicht existierenden Formats angegeben hat, so geht CFS nach Ausgabe einer entsprechenden Fehlermeldung in den formatfreien Erfassungsmodus.

Falls vom Systemverwalter ein Standardformat zum Erfassen von Dokumentationen vorgegeben ist, der Benutzer aber explizit seinen Text ohne Format erfassen möchte, so kann dies durch Angabe eines nicht existierenden Formatnamens *f* im Action-Code IM/IA erreicht werden.

Mit den Kommando CAP/LOW kann die Umsetzung von Kleinbuchstaben beim Erfassen/Ändern der Dokumentationstexte (IM) beeinflußt werden.
Standard: CAP, d.h. eingegebene Kleinbuchstaben werden in Großbuchstaben umgewandelt.

Mit dem Kommando I FORM=*f* kann ein benutzerindividuelles Format definiert werden, welches standardmäßig bei Neuerfassung von Dokumentationen verwendet wird (Action-Codes IM/IA ohne explizite Formatangabe).

Mit der User Option INF lassen sich alle Datenobjekte auffinden, für die Dokumentationen existieren/nicht existieren.

Selektionsmaske, Feld USER OPTION:

INF Zeigt an, ob für die Datei/Jobvariable bzw. für das Bibliothekselement eine Beschreibung im CFS-Informationskatalog vorliegt. Ggf. wird auch der Name des für die Erfassung verwendeten Formats angezeigt.

Selektionsparameter: [Y | N | S] [, FORM=*xxx*] [, '*string*']

Y | N	Inf-Text vorhanden ? Yes/No
S	Wie Y. Zusätzlich werden die ersten 15 Byte des Informationstextes in der Dateienliste angezeigt.
FORM=*xxx*	Auswahl bzgl. des bei der Erfassung verwendeten Formats
'*string*'	Auswahl aller Datenobjekte, die in ihrem Dokumentationstext den angegebenen Suchstring enthalten.

Bei der Angabe des Parameters FORM= bzw. bei der Auswahl nach einem Suchbegriff wird der Selektionsparameter 'Y' intern ergänzt und muß daher vom Benutzer nicht angegeben werden. Es kann mehr als ein Selektionsparameter angegeben werden. In diesem Fall sind die einzelnen Parameter durch Kommas zu trennen.

Formate in zentralen INF-Formatkatalog einbringen

Es ist die Aufgabe des Systemverwalters, die zur Erfassung von Dokumentationstexten benötigten Formate in den INF-Formatkatalog von CFS (ISAM-Datei) zu übertragen. Um dies zu erreichen, wird folgende Vorgehensweise vorgeschlagen:

1) Erstellen der gewünschten Formate mit IFG/FORMPLAG.

2) Eröffnen eines CFS-Nebenprozeßes und Einschalten des Loggings nur für die Ausgaben: LOG datei,O

3) Aufruf des IFG und Auswahl der Funktion 02: 'Format anzeigen' (Unterfunktion: 'Format wie bei seinem Einsatz') bzw. Aufruf des FORMPLAG. ETH und Aufruf der Testfunktion: .T (Maske anzeigen).

4) Beenden des IFG bzw. FORMPLAG.ETH und Ausschalten des Loggings: NLOG.

5) Anzeige der erzeugten LOG-Datei im CFS-Display: I *datei*,D. Einschalten des Modify-Modus durch Kommando M.

6) Löschen aller Datensätze in der LOG-Datei mit Ausnahme des protokollierten IFG-/FORMPLAG-Formats. Das Löschen der Datensätze erfolgt durch Überschreiben der ersten Stelle des ISAM-Schlüssels mit dem Zeichen 'E' (Eliminate Record).

7) Übertragen des einzigen, in der LOG-Datei verbliebenen Datensatzes in den zentralen Formatkatalog. Hierzu existiert in CFS ein eigenes Kommando:

(ADDFORM) *datei* [,*f*] [,*format-kat*]

datei Name der LOG-Datei, in der das gewünschte Ausgabeformat protokolliert wurde. Die Datei darf nur einen Datensatz enthalten.

f ein- bis dreistelliger Formatname, unter dem die Maske im Formatkatalog gespeichert werden soll.

format-kat Name des Katalogs, in den das aktuelle Format eingebracht werden soll. Falls der Name des Formatkatalogs nicht angegeben wurde, so wird der in CFS enthaltene Name des zentralen Formatkatalogs ($TSOS.CFS.INF.FORMAT) verwendet. Der Name des zentralen Formatkatalogs kann im Modul CFSMAIN an die aktuellen Gegebenheiten des RZs angepaßt werden.

INF-Katalog in EDT-Datei schreiben (Export)

FC2 EXP: *katalog* , *datei*

Der INF-Katalog mit den zu den Datenobjekten gespeicherten Dokumentationen kann nur von CFS verarbeitet werden, da die darin enthaltenen Datensätze eine Länge von bis zu 2048 Bytes haben. Die Schlüssel der ISAM-Datei sind 44 Stellen lang.

Mit dem Kommando FC2 EXP:... kann der Benutzer den gesamten Inhalt des Datenkatalogs in eine normale EDT-Datei exportieren. Diese Datei kann z.B. im EDT mit @ON&C'...'T'...' nach bestimmten Regeln bearbeitet werden.
Die exportierte EDT-Datei kann aber auch einfach ausgedruckt werden.

katalog Name des zu exportierenden CFS-Datenkatalogs (CFS.INF bzw. *bibliothek*. INF).

datei Name der SAM-Datei (EDT-Format), in die der Datenkatalog exportiert werden soll.

INF-Katalog aus EDT-Datei erzeugen (Import)

FC2 IMP: *datei* , *katalog* [, MERGE]

Das Kommando FC2 IMP:... bildet das Gegenstück zum oben beschriebenen Export-Kommando. Exportierte und im EDT bearbeitete INF-Kataloge können so wieder in das CFS-interne Format umgewandelt werden.

datei Name der SAM-Datei mit den Informationstexten für den CFS-Datenkatalog.

katalog Name des zu erzeugenden Datenkatalogs. (CFS.INF bzw. *bibliothek*.INF).

MERGE Die in der SAM-Datei enthaltenen Daten werden in einen bereits bestehenden Datenkatalog eingemischt.
Standard: Der Datenkatalog wird neu erstellt.

Hinweise:

Der Benutzer kann einen exportierten INF-Katalog im EDT bearbeiten und anschließend wieder importieren. Beim Importieren werden alle Informationen des alten INF-Katalogs überschrieben. Beim Editieren des exportierten Katalogs (SAM-Datei) sind folgende Regeln zu beachten:

Soll im Editor eine zusätzliche Dokumentation eingefügt werden, z.B. zur Beschreibung der Datei ABC.XYZ, so sind die folgenden zwei Datensätze einzufügen:

```
$NAME = ABC.XYZ
$
```

Es folgen die Datensätze, in denen die Dokumentationen abgelegt sind. Erstrecken sich die Dokumentationssätze über mehr als 22 Zeilen, so kann zwischen dem zweiten bis 22. Satz ein Maskenwechsel für die Action-Codes I/IM programmiert werden. Dazu ist vor dem Dokumentationssatz, der auf einer neuen Bildschirmseite erscheinen soll, ein Steuerungssatz der Form
'$$' einzufügen.

Konvertieren INF-Katalog <---> PLAM-Bibliothek

FC2 CON: *datei1* [, *datei2*] [, P]

Konvertierung ISAM --> PLAM

datei1 CFS-Informationskatalog (ISAM-Datei).

datei2 PLAM-Bibliothek (Ausgabe).
Falls *datei2* nicht angegeben wurde, so wird eine temporäre Bibliothek erzeugt, die am Ende den Informationskatalog *datei1* überschreibt.

Der Inhalt des Informationskatalogs wird in eine bereits existierende oder neu anzulegende PLAM-Bibliothek *datei2* übertragen. Für jeweils eine Dateibeschreibung wird ein Element in der Bibliothek *datei2* erzeugt: S/*xxxx*. Falls sich die Dateibeschreibung über mehrere Bildschirmseiten erstreckt (mehrere ISAM-Datensätze), so werden die entsprechenden Daten im erzeugten Bibliothekselement durch Sätze mit dem Inhalt $NEW PAGE getrennt.

Konvertierung PLAM --> ISAM

datei1 PLAM-Bibliothek.

datei2 CFS-Informationskatalog (Ausgabe). Falls *datei2* nicht angegeben wurde, so wird ein temporärer Informationskatalog erzeugt, der am Ende die PLAM-Bibliothek *datei1* überschreibt.

Der Inhalt der Elemente der PLAM-Bibliothek *datei1* wird in einen bestehenden bzw. neu anzulegenden Informationskatalog *datei2* übertragen. Zu einem Bibliothekselement *t*/*xxxxx* werden ein oder mehrere ISAM-Sätze mit Key *xxxxx* im Informationskatalog *datei2* erzeugt. Mehrere Sätze werden geschrieben, falls in den Daten des PLAM-Elements die Trennsätze $NEW PAGE enthalten sind.

P Dieser optionale Zusatz ist nur bei Konvertierung von PLAM nach ISAM von Bedeutung. Die Typbezeichnungen der Elemente in der PLAM-Bibliothek werden in die ISAM-Keys der Sätze des Informationskatalogs mit übernommen. Der so erzeugte INF-Katalog kann ausschließlich als Dokumentation für Elemente einer PLAM-Bibliothek verwendet werden.

Ausdrucken eines INF-Katalogs

Ein Informationskatalog kann als exportierte SAM-Datei mit einem Print-Kommando (ohne SPACE=E) ausgedruckt werden.

Eine andere Möglichkeit, die in einem INF-Katalog enthaltenen Informationen auszugeben, bietet die folgende Vorgehensweise:

1) Selektion aller Datenobjekte, für die Informationen im Katalog vorliegen, über die USER OPTION : INF Y.

2) Erzeugen einer Dokumentationsdatei der selektierten Datenobjekte, in die die vollständigen Informationstexte eingemischt werden. Dies ist möglich mit dem Kommando DOC *datei*,INF,ALL;A

   Die mit DOC erzeugte Datei kann mit SPACE=E ausgedruckt werden.

INF-Katalog reorganisieren

ONXICOPY *cat1*, *cat2* [,UPD]

Für die mit X angekreuzten Datenobjekte werden die Dokumentationen aus dem Katalog *cat1* in den Katalog *cat2* kopiert.

UPD *cat2* wird mit OPEN=INOUT statt mit OPEN=OUTPUT (Standardfall) eröffnet.

Die Variable Action ONXICOPY ... eignet sich zum Reorganisieren von INF-Katalogen. Oft sind in den INF-Katalogen noch Dokumentationen für schon gelöschte Datenobjekte enthalten. Die aktuellen Datenobjekte, für die im INF-Katalog Dokumentationstexte existieren, werden mit der User Option INF gezielt ausgewählt (INF=Y). Auf die so ermittelten Datenobjekte wird die Variable Action ON&ICOPY *cat1,cat2* angewendet. Die Dokumentationen für alle noch aktuellen (existierenden) Datenobjekte werden damit in einen neuen INF-Katalog übertragen. Der alte Katalog (z.B. CFS. INF) kann anschließend mit dem neuen Katalog überschrieben werden.

Auffangen der Dokumentationen für gelöschte Datenobjekte

Der ursprüngliche, noch nicht reorganisierte INF-Katalog wird mit dem Kommando FC2 EXP:*katalog,datei-a* in eine SAM-Datei exportiert. Anschließend wird der alte Katalog reorganisiert (Vorgehensweise siehe oben). Der reorganisierte Katalog wird danach ebenfalls exportiert (FC2 EXP:*katalog*, *datei-n*). Die zwei Export-Dateien werden nun in CFS mit dem Kommando COMP *datei-a*, *datei-n* verglichen. Das Differenzprotokoll enthält die Dokumentationstexte zu allen nicht mehr existierenden Datenobjekten.

14. Terminverwaltungssystem

Das Terminverwaltungssystem von CFS wird über Kommandos gesteuert und bietet dem Benutzer folgende Möglichkeiten:

1) Formatierte Anzeige eines Monatskalenders für den Zeitraum von Januar 1600 bis Dezember 2300. Vorgemerkte Termine sind im Monatskalender vermerkt.

2) Eingabe von Terminen mit Datum, Uhrzeit, Kurzinfo und ausführlichem Text. Auch bis zu 250 jährlich, monatlich oder wöchentlich wiederkehrende Termine sind möglich.

3) Änderung der Texte zu vorhandenen Terminen.

4) Anzeige aller Informationen zu einem Termin.

5) Ausgabe einer Terminliste mit Datum, Uhrzeit, Bearbeitungsvermerk und Kurzinfo zu allen Terminen bis zu einem bestimmten Datum bzw. ab einem bestimmten Datum.

6) Information in der Systemzeile des Bildschirms über die Tage, für die Termine vorgemerkt sind.

Ein Termin wird im folgenden als "Item" bezeichnet.

Den Funktionen 1) bis 6) des Terminverwaltungssystems entsprechen folgende Kommandos:

CAL Show Calendar (Monatskalender)

II Insert Item (Termin eingeben)

MI Modify Item (Text zu Termin ändern)

SI Show Item (alle Informationen zu Termin ausgeben)

SIL Show Item List (Terminliste)

SIS Kurzinformation über die Tage, für die Termine vorhanden sind.

IFILE Festlegen einer Nichtstandard-Termindatei.

Monatskalender anzeigen

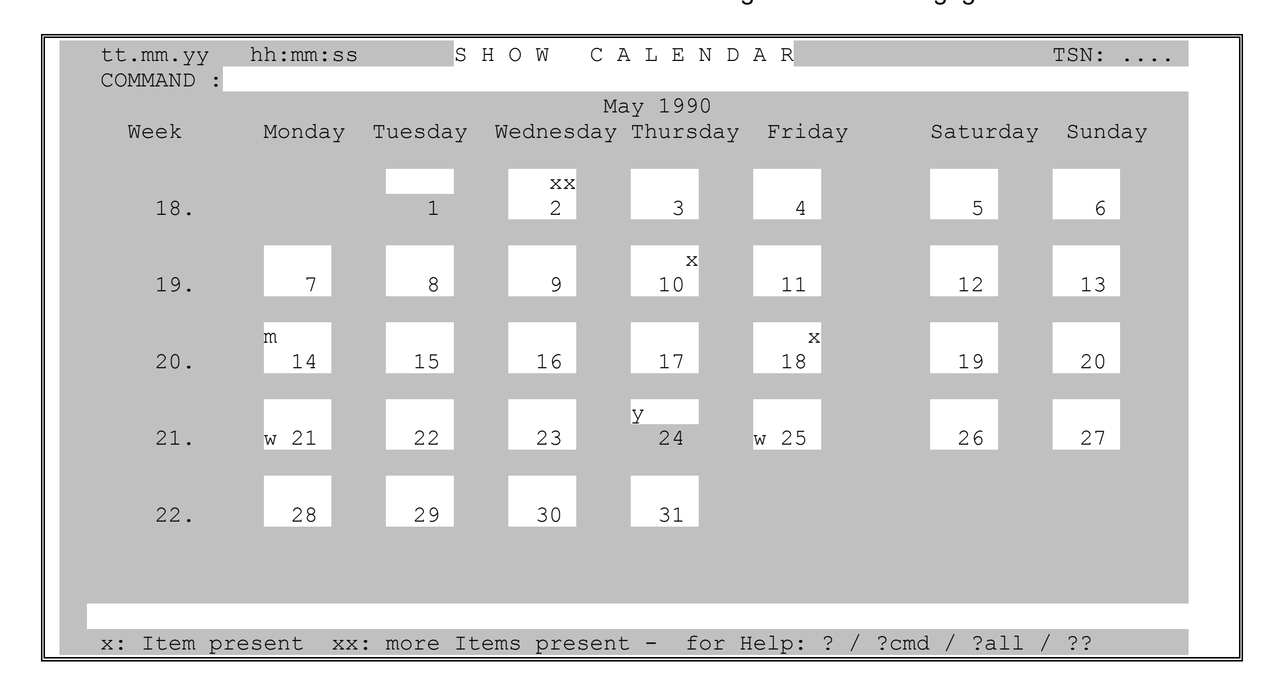
CAL [*mm.* [*yyyy*]] Calendar. Am Bildschirm wird der Kalender des angegebenen Monats angezeigt. Sind für einen Tag des angezeigten Monats Termine vorhanden, so wird bei dem betreffenden Tag ein "x" eingetragen (ein Termin), bzw. "xx" (mehrere Termine). Sind wöchentliche, monatliche bzw. jährliche Termine vorhanden, so wird ein "w", "m" bzw. "y" eingetragen. Bei der Anzeige des aktuellen Monats (CAL) wird der aktuelle Tag blinkend dargestellt.

mm. 1 <= *mm* <= 12: ein- bzw. zweistellige Monatsangabe mit Punkt am Ende.

yyyy 1600 <= *yyyy* <= 2300: Angabe des Jahres. Das Jahr kann auch als zweistellige Zahl *yy* angegeben werden. Hierbei gilt: *yy* >= 80: 19*yy*, *yy* < 80: 20*yy*.

Standard: Bei Fehlen der Monats- und Jahresangabe (Kommando CAL ohne Parameter) wird der aktuelle Monat des gegenwärtigen Jahres angezeigt. Bei Fehlen der Jahresangabe (CAL *mm.*) wird der angegebene Monat des aktuellen Jahres angezeigt.

Der Monatskalender wird am Bildschirm in folgender Form ausgegeben:



x/xx: ein bzw. mehrere Termine zu dem bezeichneten Tag
w/m/y: wöchentlich/monatlich/jährlich wiederkehrender Termin
Termine mit gesetztem Bearbeitungsvermerk (siehe Seite **436**) werden im Kalender nicht angezeigt.
Unterstrichener Tag = Feiertag

Eingabemöglichkeiten im Kommandofeld des Kalenders:

+ | - einen Monat vorwärts/rückwärts blättern.
Das Absenden des leeren Kommandofeldes mit ENTER wirkt wie "+".

CAL/II/MI/SI/SIL/SIS
Kommandos des Terminverwaltungssystems: Calendar/Insert Item/ Modify Item/ Show Item/Show Item List/Show Item Short. Ausführliche Beschreibung siehe unten.

HC | NHC Hardcopy ein-/ausschalten.

/cmd beliebiges BS2000-Kommando.

n/OC*n*/CC*n*/DC*n*/CD

beliebiges Connection bezogenes Kommando. Eine vollständige Beschreibung der einzelnen Kommandos finden Sie im Kapitel 9 "Connections" ab Seite **337**.

=*arith* Arithmetische Rechnung bzw. Hexadezimal <--> Character Konvertierung.
Ausführliche Beschreibung siehe Seite **212**.

K1 Rückkehr aus dem Kalender in die vorhergehende CFS-Maske.

F3 Aufzeichnung der aktuellen Kalendermaske in die Hardcopy-Datei von CFS.

Beispiel:

Die Ausgabe des Kalenders für Mai 1990 erfolgt durch das Kommando
CAL 5.1990 bzw. CAL 5.90

Hinweise:

Da auch im Monatskalender vorhandene Termine angezeigt werden, ist beim CAL-Kommando ein Zugriff auf die Item-Datei des Terminverwaltungssystems notwendig. Die Regel, nach der der Name der verwendeten Item-Datei ermittelt wird, ist beim Kommando IFILE weiter unten beschrieben.

Im Monatskalender werden Feiertage unterstrichen dargestellt. Der Benutzer kann darüber hinaus zusätzliche Feiertage definieren bzw. von CFS vorgegebene Feiertage als Nicht-Feiertage deklarieren. Die Definition der benutzereigenen Feiertage bzw. Nicht-Feiertage erfolgt in den sog. IFREE-Dateien (Item-Free).
Eine Feiertagsdatei CFS.IFREE [*.logon-name*|*.account*] (Fcbtype=SAM) ist in der eigenen Benutzerkennung anzulegen. Die Datei kann auch mit /FILE *datei*,LINK= IFREE zugewiesen werden. Zusätzlich zu der benutzereigenen IFREE-Datei kann eine zentrale Datei CFS.IFREE unter der CFS-Benutzerkennung angelegt werden. In diesem Fall werden die in der benutzereigenen und der zentralen IFREE-Datei definierten Feiertage gemischt.
Die in den Feiertagsdateien enthaltenen Datensätze sind in folgendem Format anzugeben: ttmm (tt = Tag, mm = Monat).
Es sind auch Eintragungen im Format ttmm- möglich. Dies bedeutet, daß der betreffende Tag nicht als Feiertag gelten soll.

Insert Item: Neuen Termin eintragen

II [*datum* [*/time*]] Insert Item. Es wird eine Leermaske zur Eingabe der Kurz- und Langinformationen für den Termin angezeigt.

datum *tt.mm.* [*yyyy*]
Angabe des Termins in Datumsform. Sowohl der Tag als auch der Monat können als ein- bzw. zweistellige Zahlen angegeben werden.
yyyy: 1600 <= *yyyy* <= 2300. Das Jahr kann sowohl als vierstellige, als auch als zweistellige Zahl angegeben werden. Falls die Jahresangabe weggelassen wurde (II *tt.mm.*), so wird das Datum um das aktuelle Jahr ergänzt.

*tt.mm.*Y Erfassen eines jährlich wiederkehrenden Termins zum Datum *tt.mm.* Das Zeichen 'Y' ist als Platzhalter für das Jahr anzugeben.

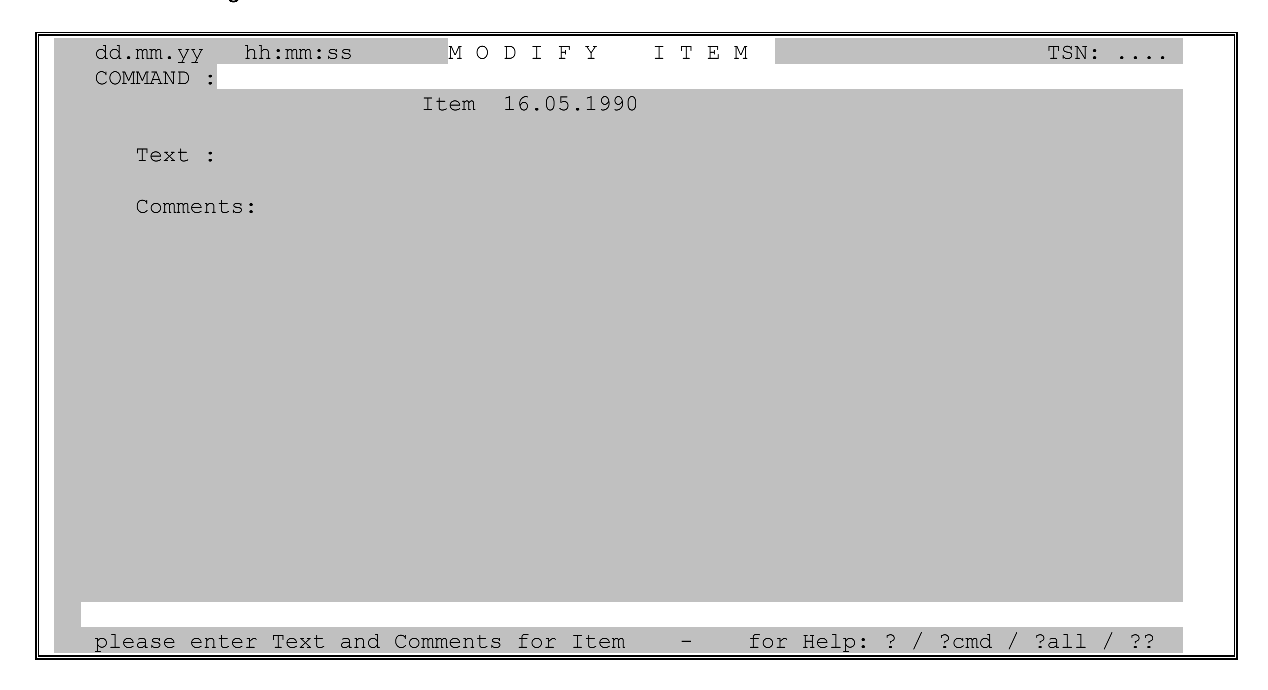
*tt.*M.Y Erfassen eines monatlich wiederkehrenden Termins zum Tag *tt.*

day Erfassen eines wöchentlich wiederkehrenden Termins.
Für *day* kann ein Wochentag angegeben werden: MON/TUE/WED/ THU/FRI/SAT/SUN bzw. MO/DI/MI/DO/FR/SA /SO

D.M.Y Erfassen eines täglich wiederkehrenden Termins.
Das spätere Löschen eines täglich wiederkehrenden Termins ist folgendermaßen möglich: Anzeige der Liste aller täglichen Termine durch SIL D.M.Y und Eintragen von E in der Markierungsspalte des zu löschenden Termins.

/time Uhrzeit für den Termin in der Form *hh:mm* oder *hh.mm*
Standard: Falls im II-Kommando kein Datum angegeben wurde, so wird das aktuelle Datum angenommen.

Aufgrund des Kommandos II wird folgende Maske zur Erfassung des Termins ausgegeben:



```
dd.mm.yy   hh:mm:ss         M O D I F Y   I T E M                    TSN: ....
COMMAND :
                          Item  16.05.1990

   Text :

   Comments:




please enter Text and Comments for Item    -    for Help: ? / ?cmd / ?all / ??
```

Im Feld "Text :" der Maske kann eine bis zu 50-stellige Kurzbeschreibung für den Termin eingegeben werden. Diese Kurzbeschreibung wird in der Terminliste (Kommando SIL, Show Item List) angezeigt.

Bei "Comments" kann eine Langbeschreibung für den Termin eingegeben werden.

Eingabemöglichkeiten im Kommandofeld der II-Maske:

CAL/II/MI/SI/SIL/SIS
Kommandos des Terminverwaltungssystems: Calendar/Insert Item/ Modify Item/ Show Item/Show Item List/Show Item Short.

HC | NHC Hardcopy ein-/ausschalten.

/cmd beliebiges BS2000-Kommando.

n/OC*n*/CC*n*/DC*n*/CD

beliebiges Connection bezogenes Kommando. Eine vollständige Beschreibung der einzelnen Kommandos finden Sie im Kapitel 9 "Connections" ab Seite **337**.

=*arith* Arithmetische Rechnung bzw. Hexadezimal <--> Character Konvertierung.
Ausführliche Beschreibung siehe Seite **212**.

K1 Rückkehr in die vorhergehende CFS-Maske.

F3 Aufzeichnung der aktuellen Maske in die Hardcopy-Datei von CFS.

Hinweis:

Die Regel, nach der der Name der verwendeten Item-Datei ermittelt wird, ist am Ende dieses Kapitels beschrieben (Kommando IFILE).

Modify Item: Text zu einem Termin ändern

MI [*datum* [*/time*]] Modify Item. Es wird der bisher eingegebene Text zu dem angegebenen Termin in einer Maske angezeigt und zur Änderung freigegeben.

datum *tt.mm.* [*yyyy*]
Angabe des Termins in Datumsform. Sowohl der Tag als auch der Monat können als ein- bzw. zweistellige Zahlen angegeben werden.
yyyy: 1600 <= *yyyy* <= 2300. Das Jahr kann sowohl als vierstellige, als auch als zweistellige Zahl angegeben werden. Falls die Jahresangabe weggelassen wurde (MI *tt.mm.*), so wird das Datum um das aktuelle Jahr ergänzt.

*tt.mm.*Y Ändern des Textes für einen jährlich wiederkehrenden Termin zum Datum *tt.mm*. Das Zeichen 'Y' ist als Platzhalter für das Jahr anzugeben.

tt.M.Y Ändern des Textes für einen monatlich wiederkehrenden Termin zum Tag *tt*.

day Ändern eines wöchentlich wiederkehrenden Termins.
Für *day* kann ein Wochentag angegeben werden: MON/TUE/WED/THU/FRI/SAT/SUN bzw. MO/DI/MI/DO/FR/SA/SO

D.M.Y Ändern des Textes zu einem täglich wiederkehrenden Termin.

/time Uhrzeit für den Termin in der Form *hh:mm* oder *hh.mm*

Standard: Falls im MI-Kommando kein Datum angegeben wurde, so wird der erste angelegte Termin zum aktuellen Datum zum Ändern dargestellt.

Das Format der MI-Maske ist bei dem Kommando II (Insert Item, siehe oben) dargestellt.

Eingabemöglichkeiten im Kommandofeld der MI-Maske:

M+ | **M-** Änderung des nächsten bzw. des vorhergehenden Termins.

CAL/II/MI/SI/SIL/SIS
Kommandos des Terminverwaltungssystems: Calendar/Insert Item/ Modify Item/ Show Item/Show Item List/Show Item Short.

HC | **NHC** Hardcopy ein-/ausschalten.

/cmd beliebiges BS2000-Kommando.

n/OC*n*/CC*n*/DC*n*/CD
beliebiges Connection bezogenes Kommando. Eine vollständige Beschreibung der einzelnen Kommandos finden Sie im Kapitel 9 "Connections" ab Seite **337**.

=*arith* Arithmetische Rechnung bzw. Hexadezimal <--> Character Konvertierung.
Ausführliche Beschreibung siehe Seite **212**.

K1 Rückkehr in die vorhergehende CFS-Maske.

 Aufzeichnung der aktuellen Maske in die Hardcopy-Datei von CFS.

Hinweise:

Das Datum oder die Uhrzeit eines bestehenden Termins können nur über die Eingabe C (Change Date) in der SIL-Maske geändert werden (siehe 436).

Sind mehrere Termine zu einem Datum vorhanden, so wird durch MI *datum* der erste dieser Termine zur Änderung angeboten. Durch das Kommando M+ kann der nächste Termin zu dem angegebenen Datum modifiziert werden.

Show Item: Text zu Termin anzeigen

SI [*datum* [*/time*]] Show Item. Es werden die ausführlichen Informationen zu dem gewünschten Termin angezeigt.

datum *tt.mm.* [*yyyy*]
Angabe des Termins in Datumsform. Sowohl der Tag als auch der Monat können als ein- bzw. zweistellige Zahlen angegeben werden.
yyyy: 1600 <= *yyyy* <= 2300. Das Jahr kann sowohl als vierstellige, als auch als zweistellige Zahl angegeben werden. Falls die Jahresangabe weggelassen wurde (*tt.mm.*), so wird das Datum um das aktuelle Jahr ergänzt.

*tt.mm.*Y Anzeigen des Textes zu einem jährlich wiederkehrenden Termin zum Datum *tt.mm.* Das Zeichen 'Y' ist als Platzhalter für das Jahr anzugeben.

*tt.*M.Y Anzeigen des Textes zu einem monatlich wiederkehrenden Termin zum Tag *tt*. Die Zeichen 'M' und 'Y' sind Platzhalter für Monat und Jahr und können auch als Kleinbuchstaben und in der Form dd.mm.yy eingegeben werden.

day Anzeigen eines wöchentlich wiederkehrenden Termins.
Für *day* kann ein Wochentag angegeben werden: MON/TUE/WED/THU/FRI/SAT/SUN bzw. MO/DI/MI/DO/FR/SA /SO

D.M.Y Anzeigen eines täglich wiederkehrenden Termins.

/time Uhrzeit für den Termin in der Form *hh:mm* oder *hh.mm*
Standard: Falls im SI-Kommando kein Datum angegeben wurde, so wird das aktuelle Datum angenommen.

Eingabemöglichkeiten im Kommandofeld der SI-Maske:

+ | - zum nächsten/vorhergehenden Termin blättern. Das Absenden des leeren Kommandofeldes hat die gleiche Wirkung wie +.

M Modify: Ändern der Textinformationen zum angezeigten Termin.

CAL/II/MI/SI/SIL/SIS
Kommandos des Terminverwaltungssystems: Calendar/Insert Item/ Modify Item/ Show Item/Show Item List/Show Item Short.

HC | NHC Hardcopy ein-/ausschalten.

/cmd beliebiges BS2000-Kommando.

n/OC*n*/CC*n*/DC*n*/CD

beliebiges Connection bezogenes Kommando. Eine vollständige Beschreibung der einzelnen Kommandos finden Sie im Kapitel 9 "Connections" ab Seite 337.

=*arith* Arithmetische Rechnung bzw. Hexadezimal <--> Character Konvertierung. Ausführliche Beschreibung siehe Seite 212.

K1 Rückkehr in die vorhergehende CFS-Maske.

K3 Rückkehr in die letzte SIL-Maske, falls dort der aktuelle Termin mit S bzw. M markiert wurde.

F3 Aufzeichnung der aktuellen Maske in die Hardcopy-Datei von CFS.

Hinweise:

Das Kommando SI zeigt den ersten, zu dem angegebenen Datum gesetzten Termin an. Weitere Termine zum gleichen Datum werden durch Absenden des leeren Kommandofeldes oder durch Eintragen von "+" im Kommandofeld erreicht.

Mit dem Kommando SILA- werden auch bereits erledigte, d.h. auf OK gesetzte Termine in einer Übersichtsliste angezeigt. Durch Eintragen des Kürzels S (Show) bei einem bereits als erledigt gekennzeichneten Termin in der SILA-Maske wird u.a. das Datum angezeigt, an dem der Termin auf erledigt gesetzt wurde.

Die Regel, nach der der Name der verwendeten Item-Datei ermittelt wird, ist auf Seite 438 beschrieben.

Show Item List: Terminliste anzeigen

SIL *datum* [+|-] Show Item List. Es werden die Kurzinformationen zu allen Terminen zu dem angegebenen Datum/bis zu dem angegebenen Datum/ab dem angegebenen Datum angezeigt.

SILA *datum* [+|-] Show Item List All. Bereits erledigte Termine (State=ok) werden bei SIL nicht angezeigt. Falls auch diese Termine in der Liste angezeigt werden sollen, so ist anstelle SIL das Kommando SILA anzugeben.

datum *tt.mm.* [*yyyy*]
Angabe des Termins in Datumsform. Sowohl der Tag, als auch der Monat können als ein- bzw. zweistellige Zahlen angegeben werden.
yyyy: 1600 <= *yyyy* <= 2300. Das Jahr kann sowohl als vierstellige, als auch als zweistellige Zahl angegeben werden. Falls die Jahresangabe weggelassen wurde (*tt.mm.*), so wird das Datum um das aktuelle Jahr ergänzt.

D.M.Y Anzeigen aller täglich wiederkehrenden Termine.
Ein Hinweis auf täglich wiederkehrende Termine wird in der Systemzeile ausgegeben.

**.*mm*.Y Anzeigen aller jährlich wiederkehrender Termine zum angegebenen Monat *mm*.

**.M.Y Anzeigen aller monatlich wiederkehrenden Termine.

	..Y	Anzeigen aller jährlich wiederkehrender Termine.
	day	Anzeigen aller wöchentlich wiederkehrenden Termine für den angegebenen Tag. Für *day* kann ein Wochentag angegeben werden: MON/TUE/WED/THU/FRI/SAT/SUN bzw. MO/DI/MI/DO/FR/SA/SO. Alle wöchentlichen Termine werden angezeigt durch das Kommando SIL MO+. In der Item-Liste kann ein wöchentlicher Termin auf OK gesetzt bzw. gelöscht werden durch die Eingabe von Y bzw. E in der Markierungsspalte. Wöchentliche Termine können nicht auf einen anderen Tag verschoben werden.

Standard: Falls im SIL-Kommando kein Datum angegeben wurde, wird das aktuelle Datum gesetzt.

+ Ausgabe aller Termine ab dem angegebenen bzw. ab dem aktuellen Datum.

\- Ausgabe aller Termine bis zu dem angegebenen bzw. bis zum aktuellen Datum.

Standard: Falls hinter dem Datum der Zusatz +/- nicht angegeben wurde, so wird nur die Liste der Termine zu dem angegebenen Datum angezeigt.

SIL +|- [*n*] Show Item List mit relativer Datumsangabe

SILA +|- [*n*] Show Item List All mit relativer Datumsangabe

+ Ausgabe aller Termine ab dem aktuellen Datum bis unbegrenzt in die Zukunft.

\- **Ausgabe aller Termine**.

+*n* Ausgabe aller Termine ab dem aktuellen Datum bis *n* Tage in die Zukunft. 1 <= *n* <= 999.

-*n* Ausgabe aller Termine in dem Zeitraum ab *n* Tagen vor dem aktuellen Datum bis unbegrenzt in die Zukunft. 1 <= *n* <= 365.

Hinweise:

Bei absoluten oder relativen Datumsangaben werden stets auch die in den Zeitraum fallenden wiederkehrenden Termine angezeigt.

Bereits erledigte Termine (State=ok) werden bei SIL nicht angezeigt. Falls auch diese Termine in der Liste angezeigt werden sollen, so ist anstelle SIL das Kommando SILA (Show Item List All) anzugeben.

Auswahl aller Termine mit einem bestimmten Suchargument im Text

SIL / **SILA** *such* Es werden alle Termine ausgegeben, deren Kurz-/Langtext den im Suchargument angegebenen String enthält.
SIL: Bei der Suche werden nur die nicht erledigten Termine berücksichtigt.
SILA: Bei der Suche werden alle, auch die als bereits erledigt gekennzeichneten Termine berücksichtigt.

such [*col*] [*r*] *item*
einfaches Suchargument wie auf Seite **309** beschrieben.

such [*vk such*] [...] ..... | ( ) | %*name* | (*s-dat*)
komplexes Suchargument wie auf Seite **313** beschrieben.

Hinweis:

Groß-/Kleinschreibung spielt bei der Suche keine Rolle. Die zu suchenden Zeichenfolgen sowie die Beschreibungstexte der Termine werden intern in Großbuchstaben umgewandelt.

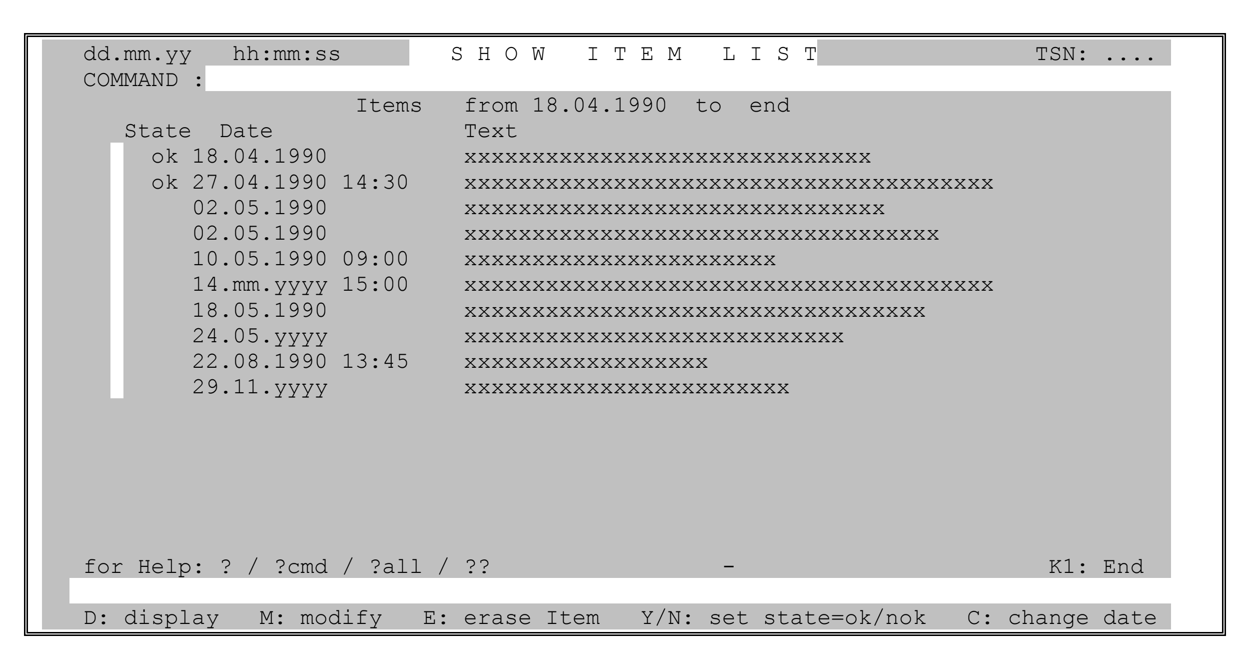
Beispiele:

`SIL 'Besprechung'`
Es werden alle Termine angezeigt, deren Kurz- bzw. Langtext die angegebene Zeichenfolge enthält.

`SIL :14:'Y'`
Es werden alle Termine mit gesetztem Bearbeitungsvermerk (State=ok, siehe Seite **436**) angezeigt.

`SIL :14:-'Y'`
Es werden alle noch nicht bearbeiteten Termine angezeigt.

Aufgrund des Kommandos SIL wird die Terminliste in folgender Form am Bildschirm ausgegeben:



```
dd.mm.yy   hh:mm:ss      S H O W   I T E M   L I S T                TSN: ....
COMMAND :
                    Items    from 18.04.1990  to  end
   State  Date               Text
     ok 18.04.1990           xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
     ok 27.04.1990 14:30     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        02.05.1990           xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        02.05.1990           xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        10.05.1990 09:00     xxxxxxxxxxxxxxxxxxxxxxx
        14.mm.yyyy 15:00     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        18.05.1990           xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
        24.05.yyyy           xxxxxxxxxxxxxxxxxxxxxxxxxxxx
        22.08.1990 13:45     xxxxxxxxxxxxxxxxxx
        29.11.yyyy           xxxxxxxxxxxxxxxxxxxxxxxx

for Help: ? / ?cmd / ?all / ??                -                       K1: End

D: display   M: modify   E: erase Item   Y/N: set state=ok/nok   C: change date
```

Monatlich bzw. jährlich wiederkehrende Termine werden in der Form *tt*.mm.yy bzw. *tt*.*mm*.yy dargestellt.

Eingabemöglichkeiten im Kommandofeld der SIL-Maske:

+ | **-** nächste / vorhergehende Seite der Terminliste.

+*n* | **-***n* in der Terminliste um *n* Zeilen vorwärts / rückwärts blättern.

++ | **--** zum Ende / Anfang der Terminliste positionieren.

SL [*datei* [,ALL]]
Save List. Die am Bildschirm angezeigte Terminliste wird in druckaufbereiteter Form in die angegebene Datei geschrieben. Der Parameter ALL bewirkt, daß neben den Kurztexten auch die Langtexte in die Datei übertragen werden.
Falls kein Dateiname angegeben wurde, so wird der Name CFS.ITEM.LIST [*.job-name*] als Standard verwendet.

CAL/II/MI/SI/SIL/SIS
Kommandos des Terminverwaltungssystems: Calendar/Insert Item/ Modify Item/ Show Item/Show Item List/Show Item Short.

HC | **NHC** Hardcopy ein-/ausschalten.

/cmd beliebiges BS2000-Kommando.

n/OC*n*/CC*n*/DC*n*/CD
beliebiges Connection bezogenes Kommando. Eine vollständige Beschreibung der einzelnen Kommandos finden Sie im Kapitel 9 "Connections" ab Seite **337**.

=*arith* Arithmetische Rechnung bzw. Hexadezimal <--> Character Konvertierung.
Ausführliche Beschreibung siehe Seite **212**.


K1 Rückkehr in die vorhergehende CFS-Maske.

F3 Aufzeichnung der aktuellen Maske in die Hardcopy-Datei von CFS.

In der **Markierungsspalte** der Terminliste sind für jeden Termin folgende Aktionen zulässig:

C Change Date/Time. Das Datum und die Uhrzeit eines Termins können hiermit geändert werden. Diese Funktion ist bei täglichen und wöchentlichen Terminen nicht zulässig.

D Display Item. Diese Eingabe bewirkt, daß neben der Kurzinformation (Text), auch die Langinformation (Comments) zu dem Termin angezeigt wird. Anstelle von D kann auch die Markierung S (Show) angegeben werden. Siehe auch Kommando SI, Seite **432**.

E Erase Item. Der Termin wird in der Item-Datei gelöscht.

I Insert (Copy Item). Ähnliche Funktion wie bei Markierung C (siehe oben). Der bisherige Termin bleibt unverändert. Es wird ein zusätzlicher Termin mit dem neu eingegebenen Datum angelegt.

M Modify Item. Analog wie D, die Informationen zum Termin werden jedoch zur Änderung freigegeben. Siehe auch Kommando MI, Seite **431**.

N State=not ok. Ein gesetzter Bearbeitungsvermerk für den Termin kann mit dieser Eingabe wieder zurück gesetzt werden.

Y State=ok. Hiermit kann ein Bearbeitungsvermerk für den Termin gesetzt werden, d.h. der Termin gilt fortan als erledigt.

Hinweis:

Die Regel, nach der der Name der verwendeten Item-Datei ermittelt wird, ist auf Seite 438 beschrieben.

Show Item Short: Kurzübersicht der Termine in der Systemzeile ausgeben

SIS *datum* [+|-] Show Item Short. In der 25. Bildschirmzeile (Systemzeile) wird das Datum zu allen Terminen im gewünschten Zeitbereich ausgegeben. Das SIS-Kommando eignet sich besonders zur Verwendung in CFS Startup-Prozeduren.

datum *tt.mm.* [*yyyy*]
Angabe des Termins in Datumsform. Sowohl der Tag, als auch der Monat können als ein- bzw. zweistellige Zahlen angegeben werden.
yyyy: 1600 <= *yyyy* <= 2300. Das Jahr kann sowohl als vierstellige, als auch als zweistellige Zahl angegeben werden. Falls die Jahresangabe weggelassen wurde (SIS *tt.mm.*), so wird das aktuelle Jahr ergänzt. Auch wiederkehrende Termine können mit SIS abgefragt werden. Näheres hierzu siehe Beschreibung des Parameters *datum* für das Kommando SIL auf Seite 433.

Standard: Falls im SIL-Kommando kein Datum angegeben wurde, so wird die Kurzübersicht der Termine zum aktuellen Datum ausgegeben.

\+ Ausgabe aller Termine ab dem angegebenen bzw. ab dem aktuellen Datum.

\- Ausgabe aller Termine bis zu dem angegebenen bzw. bis zum aktuellen Datum.

Standard: Falls hinter dem Datum der Zusatz +/- nicht angegeben wurde, so wird nur die Liste der Termine zu dem angegebenen Datum angezeigt.

SIS +|- [*n*] Relative Datumsangabe

\+ Ausgabe aller Termine ab dem aktuellen Datum bis unbegrenzt in die Zukunft.

\- Ausgabe aller Termine.

+*n* Ausgabe aller Termine ab dem aktuellen Datum bis *n* Tage in die Zukunft.
n kann eine ein- bis dreistellige Zahl sein.

-*n* Ausgabe aller Termine in dem Zeitraum ab *n* Tagen vor dem aktuellen Datum bis unbegrenzt in die Zukunft. *n* kann eine ein- bis dreistellige Zahl sein.

Aufgrund des Kommandos SIS wird in der Systemzeile des Bildschirms eine Nachricht der folgenden Art ausgegeben:

```
Items present for  18.4.90  27.4.  2.5.  10.5.*2  16.5.  13.6.  22.8.
```

Hinweis: Bei absoluten oder relativen Datumsangaben werden stets auch die in den Zeitraum fallenden wiederkehrenden Termine angezeigt.

Termindatei (Item-File) zuweisen

IFILE *datei*

Item-File. Mit diesem Kommando legt der Benutzer eine Termindatei fest.

Falls kein IFILE-Kommando angegeben wird, so wird aufgrund der folgenden Hierarchie nach einer bestehenden Item-Datei gesucht:

a) File-Kommando mit LINK=ITEM vorhanden ?

b) Jobname beim LOGON-Kommando angegeben (.name LOGON ...):
IFILE = CFS.ITEM.*name*

c) kein Jobname beim LOGON-Kommando angegeben: IFILE = CFS.ITEM

d) keine Datei mit Namen CFS.ITEM.name bzw. CFS.ITEM vorhanden:
IFILE = $*user-id*.CFS.ITEM. $*user-id* ist die Benutzerkennung, unter der das Programm CFS gespeichert ist.

e) falls keine der Bedingungen a) bis d) erfüllt ist, wird beim ersten Kommando II/MI/SI/SIS/SIL eine Item-Datei CFS.ITEM.*name* bzw. CFS.ITEM angelegt.

Anwendungsbeispiel für Terminkalender in Startup-Datei

Im folgenden Beispiel wird die Anwendung des Terminverwaltungssystems in einer Startup-Datei von CFS dargestellt. Als erstes wird in einer zentralen Geburtstagsdatei (IFILE $TSOS.CFS.ITEM.GEBURTSTAG) nachgeschaut, ob Geburtstage für heute eingetragen sind. Wenn ja, werden diese in einer Terminliste angezeigt. Danach wird die private Termindatei des Benutzers zugewiesen (IFILE CFS.ITEM. &LOGNAM). In der Übersicht werden alle Termine angezeigt, die seit dem letzten Aufruf von CFS oder in der Zukunft fällig sind. In der Systemzeile werden zusätzlich die heute aktuellen Termine ausgegeben. Zum Schluß wird die Jobvariable mit dem Datum des letzten CFS-Aufrufs aktualisiert.

```
*002 FILE=NO
* ****** *  T E R M I N K A L E N D E R   ZEIGEN  * **************
IFILE $TSOS.CFS.ITEM.GEBURTSTAG
SIL
*IF %OUT=C'S H O W   I T E M   L I S T' SKIP .GT
*SKIP .NGT
.GT
*ON %CMD CONTINUE
*DIA
.NGT
* ANZEIGE DER EINTRAEGE IM TERMINKALENDER AB DEM DATUM, AN DEM
* CFS ZULETZT GELADEN WAR
IFILE CFS.ITEM.&LOGNAM
/DCLJV CFS.JV.&LOGNAM..LAST.START
/DCLJV #X
/SETJV #X,CFS.JV.&LOGNAM..LAST.START
*ON %CMD CONTINUE
SIL &(#X) +
* IN SYSTEMZEILE NUR TERMINE VON HEUTE ANZEIGEN !
*DIA SIS
/SETJV CFS.JV.&LOGNAM..LAST.START,$SYSJV.DATUM
*  ***** *  ENDE  T E R M I N K A L E  N D E R  * ***************
*DIA
```

15. Hardcopy

Der Benutzer kann für alle innerhalb von CFS ausgegebenen Bildschirmmasken eine Protokollierung veranlassen. CFS protokolliert die Bildschirmmasken zeilen- und spaltengerecht. Außer Bildschirmmasken werden auch die ausgeführten Actions protokolliert.

Durch das Kommando HC wird die Protokollierung (Hardcopy-Modus) eingeschaltet. Von diesem Zeitpunkt an werden alle Masken und Actions des Grundprozeßes von CFS protokolliert. Der Hardcopy-Modus kann wieder ausgeschaltet werden durch das Kommando NHC (No HardCopy).

Ohne den Hardcopy-Modus explizit einzuschalten, können einzelne Masken im Basisprozeß von CFS (z.B. Ausschnitte aus der gerade angezeigten Display-Datei) auch protokolliert werden, indem sie mit der Datenübertragungstaste F3 abgesendet werden. Es ist jedoch zu beachten, daß jede mit **F3** protokollierte Maske im ausgedruckten Listing auf einer neuen Seite erscheint. Im normalen Hardcopy-Modus (HC-Kommando) werden nach Möglichkeit mehrere Masken auf einer Druckseite untergebracht. Das Absenden einer CFS-Maske mit **F3** wird, abgesehen von der Protokollierung, wie ENTER behandelt.
Über ein File-Kommando mit LINK=HARDCOPY kann eine Hardcopy-Datei mit einem vorgegebenen Namen zugewiesen werden.

Auch für alle mit OC eröffneten Connection können die Ein-/Ausgaben mitprotokolliert werden. Für eine Connection kann der Hardcopy-Modus auf zweierlei Arten aktiviert werden. Entweder innerhalb der Connection selbst durch HC [*datei*] [, *params*] oder im Grundprozeß von CFS durch HC*n*. Der Hardcopy-Modus wird ausgeschaltet durch das Kommando NHC (einzugeben innerhalb der Connection) bzw. durch NHC*n* (einzugeben im Feld COMMAND in der Grundmaske).

Das Hardcopy-Kommando mit den möglichen Optionen ist im folgenden dargestellt.

HC [*n*] [*datei*] [, L] [, B] [, T] [, E|O] [, A|F] [, G|I] [, S] [, R] [, RS] [, SMO]

Alle Optionen des HC-Kommandos sind wahlweise. Werden sie nicht angegeben, so gelten die entsprechenden Standardwerte.

n — Im Basisprozeß von CFS ist die Nummer *n* im HC-Kommando anzugeben, falls der Hardcopy für die Connection *n* gewünscht wird.

Wird im Basisprozeß von CFS keine Nummer *n* im HC-Kommando angegeben, so wird der Hardcopy-Modus für die CFS-Grundebene eingeschaltet.

Befindet sich der Benutzer in einer Connection, so muß die Nummer *n* im HC-Kommando nicht angegeben werden, um den Hardcopy-Modus für die aktuelle Connection einzuschalten.

datei — Name der Datei, in der der Dialog abgelegt werden soll. Die Hardcopy-Datei wird stets in der Benutzerkennung des Grundprozeßes von CFS erzeugt. Es können für jede Connection eigene Hardcopy-Dateien angelegt werden.

Existiert die angegebene Hardcopy-Datei bereits und wurde keine der Optionen O/E (siehe unten) angegeben, so wird der Benutzer gefragt, ob die bestehende Hardcopy-Datei erweitert oder überschrieben werden soll.

Falls der Hardcopy-Vorgang in der Connection zeitweilig ausgeschaltet wurde (NHC) und mit HC später wieder eingeschaltet wird, so wird die zuletzt eröffnete Hardcopy-Datei erweitert.

Wird beim ersten HC-Kommando innerhalb einer Connection kein Dateiname angegeben, so wird eine Hardcopy-Datei mit dem Standardnamen CFS.HARDCOPY.*yymmdd.hhmm.tsn* angelegt.

L **Laserdruck-Aufbereitung**:
Die Hardcopy-Datei wird im laserausdruckbaren Format (ND-/HPFILE) geschrieben. Helle und halbhelle Maskenfelder, geschützte und ungeschützte Maskenfelder können mit der Laser-Option auch im Druckbild entsprechend dargestellt werden.

Die mit der L-Option erzeugte Hardcopy-Datei kann mit dem BS2000 Kommando `/DO CFS.DO.PRINT,datei` ausgedruckt werden.
Die DO-Prozedur CFS.DO.PRINT, sowie die zugehörige CFS.ND-/HPFILE müssen unter TSOS zur Verfügung stehen und shareable sein. Der Systemverwalter muß die PROC-Anweisung dieser Prozedur evtl. noch bezüglich der verwendeten HP-/NDFILE und der verwendeten Zeichensätze anpassen. Hinweise hierzu sind in der Prozedurdatei enthalten.

Standard: L-Option ausgeschaltet, d.h. die Hardcopy-Datei wird im normalen Druckformat erzeugt.

B Before-Images (nur in Connections):
Bei Bildschirmmasken wird zusätzlich zur standardmäßigen Protokollierung nach der Benutzereingabe auch die Maske im Zustand vor der Eingabe protokolliert.

Standard: B-Option ausgeschaltet (keine Before-Images).

T Time-Stamps:
Die protokollierten Ein-/Ausgaben werden zusammen mit Datum und Uhrzeit in der Hardcopy-Datei abgelegt.
Standard: T-Option ausgeschaltet (kein Time-Stamping).

E | O Extend/Overwrite:

E Falls die angegebene Hardcopy-Datei bereits existiert, so wird sie im Extend-Modus eröffnet.

O Die angegebene Hardcopy-Datei wird im Output-Modus eröffnet, d.h. der bisherige Inhalt wird überschrieben. Anstelle von 'O' kann auch 'N' (New file) angegeben werden.

Standard: Falls eine Hardcopy-Datei gleichen Namens bereits existiert, wird der Benutzer gefragt, ob er die bestehende Datei überschreiben oder erweitern möchte.

A | F 1) Mit Laseraufbereitung (L-Option eingeschaltet):

A Die Anzeigesteuerzeichen der Bildschirmmasken werden auf hell und halbhell untersucht. Halbhelle Felder werden im grau hinterlegten Zeichensatz dargestellt.

F Die Feldbehandlungszeichen der Bildschirmmasken werden auf geschützte und ungeschützte Felder untersucht. Geschützte Felder werden im grau hinterlegten Zeichensatz dargestellt.

Die A- und F-Option dürfen nicht zusammen angegeben werden.

Standard: A-Option (Anzeigesteuerzeichen werden ausgewertet).

2) Ohne Laseraufbereitung (L-Option nicht eingeschaltet):

F — In der Hardcopy-Datei werden die IS2-Folgen der Bildschirmmasken mit abgelegt. Dies gibt dem Benutzer die Möglichkeit, die IS2-Folgen selbst in geeignete Umschaltzeichen für besondere Zeichensätze des Laserdruckers umzuwandeln.

A — In der Hardcopy-Datei werden die IS3-Folgen der Bildschirmmasken mit abgelegt. Dies gibt dem Benutzer die Möglichkeit, die IS3-Folgen selbst in geeignete Umschaltzeichen für besondere Zeichensätze des Laserdruckers umzuwandeln.

Die F-Option und die A-Option können auch zusammen angegeben werden. In der Hardcopy-Datei erscheinen dann IS2- und IS3-Folgen.

Standard: A-/F-Option ausgeschaltet (IS2-/IS3-Folgen werden nicht in die Hardcopy-Datei übernommen).

G | I — Nur bei eingeschalteter L-Option: German/International

G — Deutsche Umlaute werden im Laserausdruck als solche dargestellt. Dies erfolgt unabhängig von der Darstellung am Bildschirm.

I — Deutsche Umlaute werden gemäß der internationalen Zeichendarstellung in der Hardcopy-Datei als eckige/geschweifte Klammern dargestellt.

Standard: G-Option (Darstellung der deutschen Umlaute).

S — Screen. Nach Eingabe des HC-Kommandos im Line-Modus in einer Connection werden normalerweise alle Ausgaben und Eingaben in der Hardcopy-Datei festgehalten. Der Inhalt des Bildschirms zum Zeitpunkt des HC-Kommandos wird hierbei nicht erfaßt. Mit der S-Option wird nach dem Einschalten des Hardcopy-Modus der gesamte, zu diesem Zeitpunkt sichtbare Bildschirminhalt in der Hardcopy-Datei festgehalten. Die S-Option ist nur sinnvoll bei Eingabe des HC-Kommandos im Line-Modus in einer Connection.

Standard: S-Option nicht eingeschaltet.

R — Reduced. Reduzierte Ausgabe in die Hardcopy-Datei. Bei der reduzierten Ausgabe wird auf die Bildschirmrahmen verzichtet. Darüberhinaus werden nur die Bildschirmzeilen 5 bis 23 bzw. 6 bis 23 (Help-System) protokolliert. Dies hat zur Folge, daß z.B. im Display-Modus von CFS nur die am Bildschirm angezeigten Dateiinhalte in der Hardcopy-Datei erscheinen.
Der R-Modus ist nicht sinnvoll bei laseraufbereiteten Hardcopy-Dateien (siehe Parameter L) sowie beim Hardcopy von Connections.

Standard: R-Option nicht eingeschaltet.

RS — Reduced Size: Dieser Modus bewirkt beim Hardcopy von Bildschirmmasken, daß die Umrahmung der Masken unterdrückt wird. Außerdem werden in den Ausgabesätzen alle rechts stehenden Blanks gelöscht. Damit wird eine deutliche Reduzierung der Dateigröße besonders bei umfangreichen Hardcopy-Vorgängen erreicht.

SMO — Single Mask Output: Bei mehreren aufeinanderfolgenden Masken wird nur die letzte in der Hardcopy-Datei aufgezeichnet. D.h. nach jeder Maskenaufzeichnung wird die Hardcopy-Datei automatisch geschlossen und vor der nächsten Maske mit Open Output geöffnet.

NHC-Kommando

NHC | NHC*n* Hardcopy-Modus für CFS-Grundebene/für Connection *n* ausschalten.
In einem Nebenprozeß wird die Hardcopy-Datei bei LOGOFF/KDCOFF, Disconnect/ Connection Lost automatisch geschlossen.

Hinweise:

HC-Kommando in Connections.

Innerhalb einer Connection kann das HC-Kommando sowohl im Line-Modus, als auch im Format-Modus eingegeben werden. In einer Bildschirmmaske ist das HC-Kommando im ersten modifizierbaren Feld einzutragen. Eine Ausnahme hiervon bildet der EDT. Hier ist das HC-Kommando in der 24-ten Zeile (Kommandozeile) einzugeben. In der gleichen Maske darf außerdem kein weiteres Feld modifiziert werden. Dasselbe gilt für das NHC-Kommando.

Shift Reset Bild Bei Bildschirmmasken, in denen das erste modifizierbare Feld nicht zur Aufnahme des HC-Kommandos ausreicht, kann mit der Taste Shift Reset der Bildschirm gelöscht und in Zeile 1, Spalte 1 das gewünschte HC-Kommando eingegeben werden.

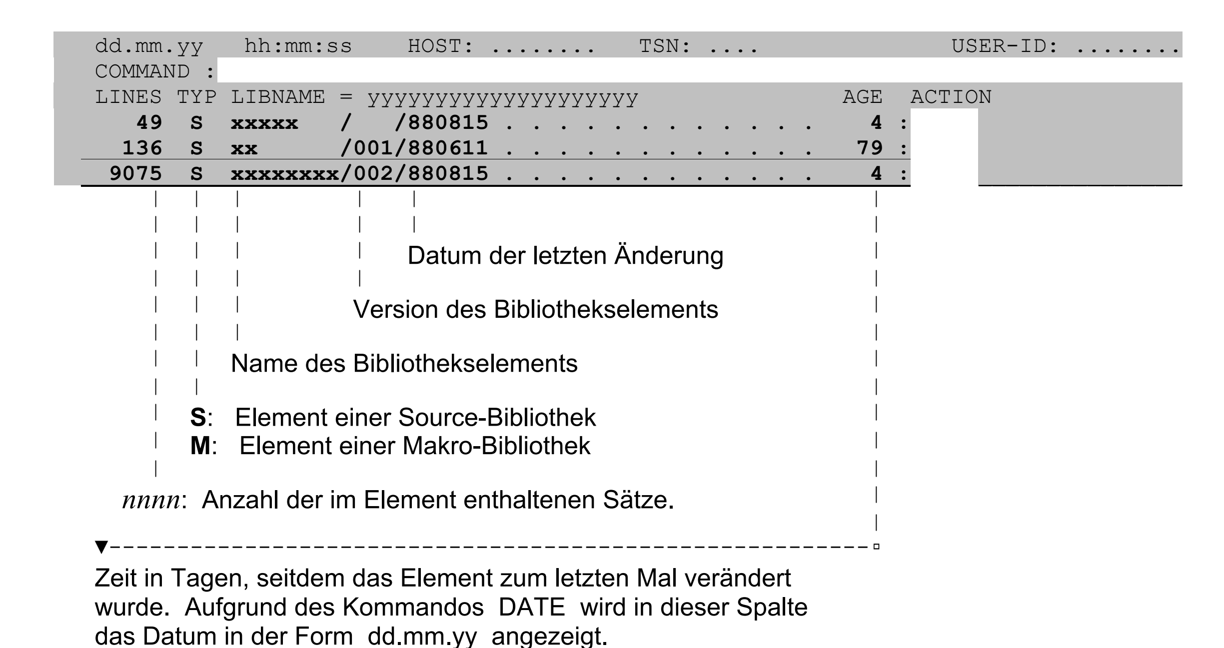
16. Bearbeitung von LMS-Bibliotheken

LMS-Bibliotheken existieren in zwei verschiedenen Ausprägungen: LMS-ISAM (OSM) und PLAM. In den folgenden Seiten wird auf die jeweiligen Besonderheiten der Bibliotheken bei der Bearbeitung durch CFS eingegangen.

Bibliotheken vom Typ LMS-ISAM (OSM)

In diesem BS1000-kompatiblen Bibliotheksformat werden Elemente durch einen ein- bis achtstelligen Elementnamen und eine dreistellige Versionskennung bezeichnet. Die Bibliotheken sind als ISAM-Dateien organisiert und werden in der vorliegenden Beschreibung auch als LMS-ISAM Bibliotheken bezeichnet. Die ebenfalls verwendete Bezeichnung OSM (Old Source Macro) rührt daher, daß in Bibliotheken dieses Typs überwiegend Quellprogramme (Sourcen) und Makros abgelegt sind.

In CFS werden Elemente einer LMS-ISAM Bibliothek in dem folgenden Format dargestellt:



Auswahl der Elemente mit den jeweils höchsten Versionen

VER Durch Angabe dieser User Option wird die Anzahl der unter dem Elementnamen gespeicherten Versionen ausgegeben. In der Dateienliste wird nur die höchste Version eines jeden Elements angezeigt. Mit dem Action-Code NP können alle Versionen des Elements selektiert werden.

Aufnahme von Elementen in eine LMS-ISAM Bibliothek (Add).

Wird bei der Aufnahme von Dateien in eine LMS-Bibliothek der Elementname nicht explizit angegeben, so wird er aus den letzten 8 Zeichen der letzten Teilqualifizierung des Dateinamens gebildet. Enthält der Dateiname ein Zeichen '#', so werden die darauf folgenden Zeichen von CFS als Versionsangabe für das LMS-Bibliothekselement interpretiert.

Es folgt eine Zusammenstellung der Kommandos, durch die die Art der Aufnahme von Dateien in eine LMS-ISAM Bibliothek gesteuert wird.

AC | NAC Add Compressed/No Add Compressed (nur für LMS-ISAM).

Nachfolgende ADD-Kommandos/Actions bewirken eine komprimierte/unkomprimierte Abspeicherung in die Bibliothek.
Standardwert: AC (komprimierte Abspeicherung).

Der NAC-Modus kann auch durch /SETSW ON=10 gesetzt werden.
(Anfangseinstellung: /SETSW OFF=10 entspricht AC-Modus).

AI | NAI Add with Increment of version/No Increment of version (nur für LMS-ISAM).

AI Die Aufnahme eines Datenobjekts (Datei/Bibliothekselement) in eine Bibliothek erfolgt in der Weise, daß die Versionsnummer gegenüber dem letzten bereits gespeicherten Element gleichen Namens um 001 erhöht wird. Falls noch kein entsprechendes Element in der Bibliothek existiert, wird ein Element mit der Version 001 erzeugt.

NAI Bei der Aufnahme eines Datenobjekts (Datei/Bibliothekselement) in eine Bibliothek wird das bereits gespeicherte Element gleichen Namens mit der höchsten Version überschrieben. Existiert noch kein Element des entsprechenden Namens, so wird ein neues Element mit der Version 001 angelegt.
Standard: NAI, d.h. es wird keine automatische Versionserhöhung vorgenommen.

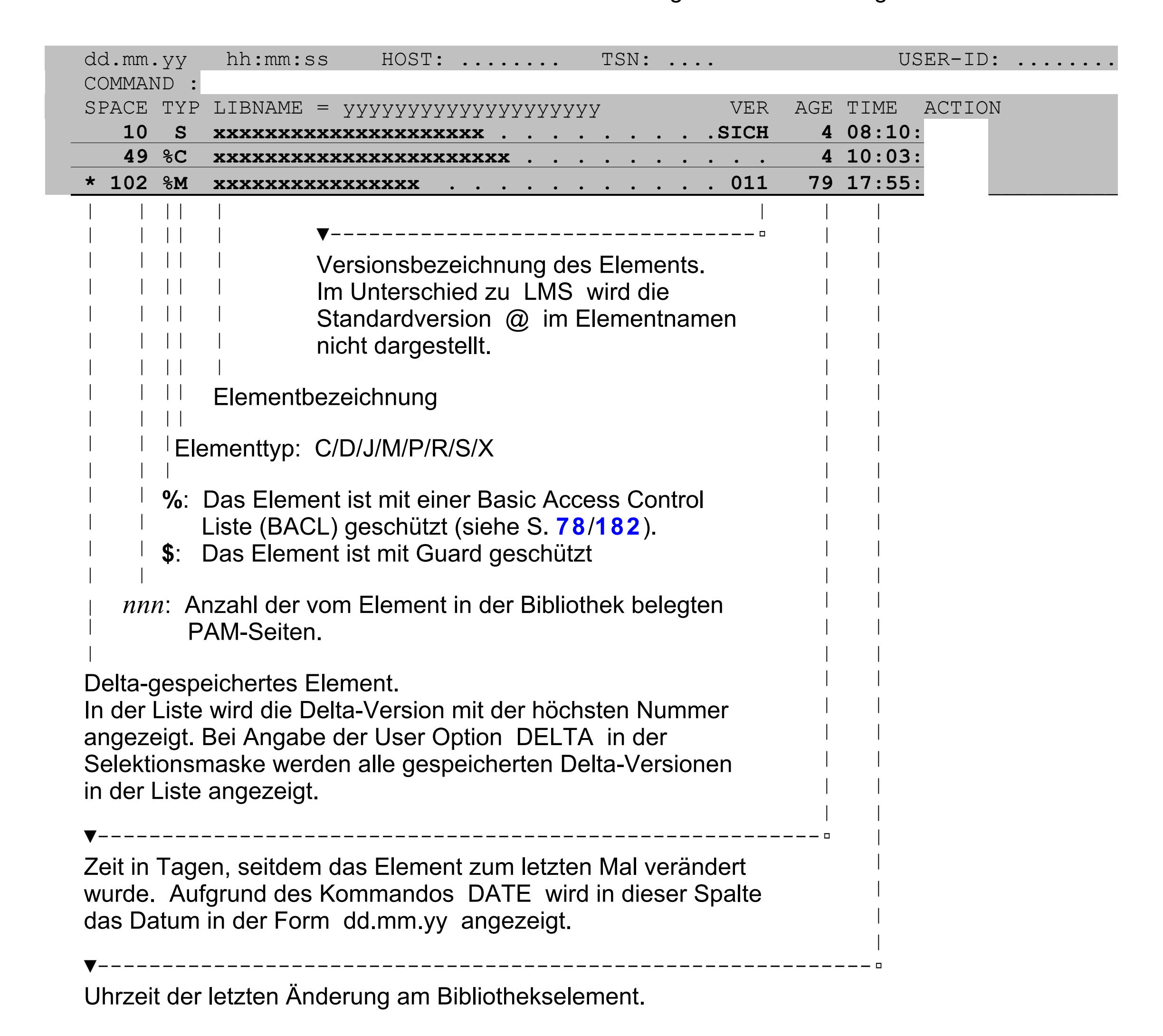
Bibliotheken vom Typ PLAM (Programmbibliotheken)

Bibliotheken dieses Formats werden auch als Programmbibliotheken bezeichnet. In PLAM-Bibliotheken können alle zu einer Programmentwicklungs-Umgebung gehörigen Datenobjekte gespeichert werden. Der Bedarf an freistehenden Dateien reduziert sich daher erheblich, der TSOSCAT wird kleiner und der Dateizugriff wird beschleunigt.

Elemente einer PLAM-Bibliothek werden bezeichnet durch:

1) Element-Typ (1 bis 8 Stellen).
2) Elementbezeichnung (1 bis 64 Stellen),
3) Versionsbezeichnung (1 bis 24 Stellen).

In CFS werden Elemente einer PLAM-Bibliothek in dem folgenden Format dargestellt:



In CFS stehen für die vollständige Bezeichnung eines PLAM-Elements 44 Stellen zur Verfügung. Diese können frei zwischen Typ-, Element- und Versionsbezeichnung aufgeteilt werden. Abweichend von LMS kann eine beliebige, bis zu acht Stellen lange Typbezeichnung verwendet werden.

Kommandos zur Steuerung der Typ-Überprüfung

CFS kann bei Aufnahme (Add) von Elementen in PLAM-Bibliotheken den Elementtyp auf LMS-Kompatibilität prüfen. Es sind zwei Alternativen möglich:

a) Nicht LMS-kompatible Typbezeichnungen werden abgewiesen

b) Der Benutzer wird gefragt, ob er wirklich ein Element mit einer nicht LMS-kompatiblen Versionsbezeichnung in der Bibliothek abspeichern möchte.

CT Check Types. Nur LMS-kompatible Elementtypen werden bei Aufnahme in PLAM-Bibliotheken zugelassen.

CTQ Check Types Query. Bei nicht LMS-kompatiblen Elementtypen wird eine Bestätigung durch den Benutzer verlangt (Standard).

NCT No Check Types. Es erfolgt keine Typ-Überprüfung durch CFS.

LMS-kompatible Elementtypen sind:

C Coreimage (mit TSOSLNK gebundene Programme)
D Document/Data
H von H-Assembler erzeugte Daten
J JCL (DO-/Enter-Prozeduren)
L vom Binder des BS2000 erzeugte Daten
M Macro
P Print (mit SPACE=E ausdruckbare Listdateien)
R Relocatable (Objektmodule)
S Source
X Extra

Auswahl aller Elemente mit einen GUARD-Zugriffsschutz

GUARD Es wird angezeigt, welche Zugriffsarten für Dateien/PLAM-Bibliothekselemente mit Guard geschützt sind (ab BS2000 V11.0).
R: Der lesende Zugriff (Read) ist mit Guard geschützt
W: Der schreibende Zugriff (Write) ist mit Guard geschützt
X: Der ausführende Zugriff (Exec) ist mit Guard geschützt

Mit den Selektionsangaben Y | N werden alle Datenobjekte ausgewählt, für die mindestens eine Zugriffsart mit Guard geschützt ist, bzw. alle, für die kein Guard-Schutz definiert ist.

Mit den Selektionsangaben [-] R|W|E [=*name*] werden alle Datenobjekte ausgewählt, für die eine bestimmte Zugriffsart [nicht] mit einem beliebigen Guard bzw. mit einem Guard des angegebenen Namens geschützt ist.

Selektionsparameter: Y | N | [-]R|W|E [=*name*]

Beispiele:

```
GUARD W
```
Es werden alle Dateien/Bibliothekselemente selektiert, die bezüglich des schreibenden Zugriffs mit einem Guard geschützt sind.

```
GUARD R=$TSOS.READGUARD
```
Es werden alle Dateien/Bibliothekselemente selektiert, die bezüglich des lesenden Zugriffs mit einem Guard des Namens $TSOS.READGUARD geschützt sind.

Auswahl aller Elemente, die von einem Benutzer ausgeliehen sind

HOLD

Hold-Status für PLAM-Elemente. Ab LMS Version 3.0 können Elemente einer Bibliothek ausgeliehen werden. Dies bedeutet, daß die Benutzerkennung des Entleihers im Inhaltsverzeichnis der Bibliothek vermerkt wird. Nur Benutzer, die unter der Kennung des Entleihers angeloggt sind, können das Element in der gleichen oder in einer höheren Version ändern oder kopieren.

Die User Option HOLD zeigt den Ausleihstatus der einzelnen Elemente. Die Anzeige 'N' bedeutet, daß das Element nicht ausgeliehen ist. Falls das Element ausgeliehen ist, wird die Benutzerkennung des Entleihers angezeigt.

Selektionsparameter: N | Y [:*user-id*] (*user-id* ohne $ und Punkt)

Durch die Selektionsangabe Y werden alle ausgeliehenen Elemente ausgewählt.
Mit Y:*user-id* werden alle Elemente ausgewählt, die unter einer bestimmten Kennung ausgeliehen wurden.

Hinweise:

Bibliothekselemente können mit dem Action-Code HOLD ausgeliehen werden. Mit NHOLD kann der Ausleihstatus zurückgesetzt werden. Für weitere Informationen hierzu siehe Seite **195**.

Für eine ausführliche Beschreibung der Auswirkungen von HOLD wird auf das LMS-Benutzerhandbuch Version 3.0 verwiesen: Kapitel 3 (Elemente), Abschnitt 3.6 (Elementschutz/Datenschutz), Thema: "Übersicht über notwendige Rechte für LMS-Aktionen".

Bibliothekselement ausleihen/Ausleihung zurücksetzen

HOLD | **NHOLD**

Ab LMS Version 3.0 können Elemente einer Bibliothek ausgeliehen werden. Dies bedeutet, daß die Benutzerkennung des Entleihers im Inhaltsverzeichnis der Bibliothek vermerkt wird. Nur Benutzer, die unter der Kennung des Entleihers angeloggt sind, können das Element in der gleichen oder in einer höheren Version ändern oder kopieren.

Mit dem Action-Code HOLD wird ein Element ausgeliehen. Hierbei ist es von Bedeutung, unter welcher Benutzerkennung die Ausleihung erfolgt.

Mit dem Action-Code NHOLD wird der Ausleihstatus für ein Element zurückgesetzt.

Anzeige der Elemente mit der jeweils höchsten Versionsnummer

VER

Es wird die Anzahl der unter dem Elementnamen gespeicherten Versionen ausgegeben. Bei Angabe der User Option VER wird nur die höchste Version eines jeden Elements in der Dateienliste angezeigt. Dies ist insbesondere auch bei nicht deltagespeicherten PLAM-Elementen der Fall. Bei deltagespeicherten PLAM-Elementen wird standardmäßig nur die höchste Version in der Dateienliste angezeigt.

Aufnahme von Elementen in eine PLAM-Bibliothek (Add).

Bei Aufnahme von Dateien wird der Elementname aus dem vollständigen Dateinamen (ohne User-Id) gebildet. Der Elementtyp sowie die Versionsbezeichnung können zusätzlich angegeben werden. Falls der Elementtyp im Add-Kommando nicht angegeben ist, wird S/ als Standard-Typ angenommen. Falls die Elementversion im Add-Kommando nicht angegeben ist, wird bei bereits existierenden Elementen dasjenige mit der höchsten Version überschrieben. Bei neu aufzunehmenden Elementen wird /ß (X'FF') als Standard-Versionsbezeichnung verwendet.

PLAM-Bibliotheken und Delta-Speicherung

Delta-Speicherung bedeutet, daß verschiedene Versionen des gleichen Elements in einer Form abgelegt sind, bei der nur die bezüglich der Vorgängerversion geänderten Sätze gespeichert werden. Durch die Delta-Technik kann im Vergleich zur Vollspeicherung der Elemente eine bedeutende Reduzierung des Speicherplatzes erreicht werden.

User Option DELTA

Die User Option DELTA erfüllt bei der Selektion von PLAM-Bibliotheken drei verschiedene und voneinander unabhängige Funktionen. Die Funktionen a) und b) sind an die bloße Angabe der User Option in der Selektionsmaske und nicht an spezielle Selektionsparameter gebunden.

a) Bei delta-gespeicherten Bibliothekselementen werden durch die Angabe der User Option alle vorhandenen Versionen mit einem eigenen Eintrag in der Dateienliste dargestellt. In Normalfall, d.h. falls die User Option nicht angegeben wurde, wird bei delta-gespeicherten Bibliothekselementen jeweils nur die höchste vorhandene Version mit einem Eintrag in der Dateienliste dargestellt.

b) Wird eine in der Dateienliste aufgeführte delta-gespeicherte Elementversion mit der User Option ONXADD in eine andere PLAM-Bibliothek übertragen, so wird das Element in Delta-Form in die neue Bibliothek übernommen. In Normalfall, d.h. ohne Angabe der User Option wird ein delta-gespeichertes Element bei ONXADD als nicht deltagespeichertes Element übertragen, falls das Element in der Zielbibliothek noch nicht existiert oder zumindest nicht als Delta-Element existiert.

c) Durch Angabe der User Option werden bei delta-gespeicherten Elementen einer PLAM-Bibliothek alle vorhandenen Delta-Versionen in der Dateienliste dargestellt. In der User Option Spalte wird für jede Element-Version die bezüglich der Delta-Bildung verwendete Basisversion angezeigt. Die erste Version eines Elements hat definitionsgemäß sich selbst als Vorgänger. In der Liste wird die erste Version durch einen Stern (*) in der User Option Spalte dargestellt.
Durch den Selektionsparameter Y werden nur die delta-gespeicherten Elemente der PLAM-Bibliothek selektiert.
Durch den Selektionsparameter N werden alle nicht delta-gespeicherten Elemente der PLAM-Bibliothek selektiert.

Aufnahme von Elementen in eine PLAM-Bibliothek: Kommando ADD / Variable Action ONXADD

Wird ein Element aus einer anderen Bibliothek oder eine Datei mit ONXADD in eine PLAM-Bibliothek übertragen, so sind bei der Delta-Speicherung verschiedene Fälle zu unterscheiden:

1) In der Variablen Action ONXADD wurde der Parameter CA (Copy All) angegeben. In diesem Fall werden bei einem deltagespeicherten Element automatisch alle Deltaversionen strukturerhaltend in die Zielbibliothek übertragen. Die folgenden Punkte 2) und 3) beziehen sich auf den Fall, daß die CA-Option nicht angegeben wurde (Standard).

2) In der Zielbibliothek existiert bereits ein Element des angegebenen Namens und Elementtyps. In diesem Fall wird die Speicherungsform des bereits bestehenden Elements beibehalten. Dies bedeutet:

   a) Es liegt ein Element als Voll-Element (d.h. in nicht delta-gespeicherter Form) vor. Das bisherige Element wird durch das neu aufzunehmende Element gleichen Namens überschrieben, sofern nicht verschiedene Versionsnummern vorliegen.

   b) Es liegt ein Element in delta-gespeicherter Form vor. Das neu aufzunehmende Element (die neu aufzunehmende Datei) wird als eine neue Version durch Ergänzung um die nächsthöhere Versionsnummer an den bestehenden Delta-Baum angehängt. Das alte Element wird dadurch nicht überschrieben und steht weiterhin zur Verfügung.

3) In der Zielbibliothek existiert kein Element des angegebenen Namens. In diesem Fall sind folgende Möglichkeiten gegeben:

   a) Das zu übertragende Element liegt in der Ursprungsbibliothek in nicht delta-gespeicherter Form vor. Das Element wird dann ebenfalls in nicht delta-gespeicherter Form in der Zielbibliothek angelegt.

   b) Das zu übertragende Element liegt in der Ursprungsbibliothek in delta-gespeicherter Form vor.

   - Das Element wird in der Zielbibliothek in nicht delta-gespeicherter Form angelegt, falls die User Option DELTA bei der Selektion der Ursprungsbibliothek nicht angegeben wurde.

   - Das Element wird in der Zielbibliothek in delta-gespeicherter Form angelegt, falls die User Option DELTA bei der Selektion der Elemente der Ursprungsbibliothek angegeben wurde.
     Die ausführliche Beschreibung der User Option DELTA finden Sie auf Seite 82.

PLAM-Bibliothekselemente in delta-gespeicherte Form konvertieren

ONX**CONV** DELTA=*vers*

Die durch den Action-Code X gekennzeichneten PLAM-Bibliothekselemente werden in Basiselemente für Delta-Speicherung angelegt (Delta Install Version).

vers 1- bis zehnstellige numerische Versionsbezeichnung des Delta-Elements.
Bei jeder nachfolgenden Modifikation des Bibliothekselements wird die Versionsbezeichnung von CFS automatisch um 1 erhöht. Es ist daher sicherzustellen, daß in der anfänglichen Version genügend viele Stellen reserviert werden. Z.B. ONXCONV DELTA=0001. Das so eingerichtete Delta-Element kann 9998 mal modifiziert werden.

Hinweise:

Es können nur solche PLAM-Elemente in Delta-Form konvertiert werden, die keine Format-B Sätze enthalten. Nicht konvertiert werden können z.B. gebundene Programmphasen (TYP=C) und andere als Bibliothekselemente archivierte PAM-Dateien.

Voraussetzung für die korrekte Bearbeitung von delta-gespeicherten Elementen einer PLAM-Bibliothek ist, daß LMS in der Version 1.4 oder höher im Rechenzentrum zur Verfügung steht.

Guard-Schutz für Bibliothekselemente festlegen

ONX**GUARD** [%*name* | %? [,JRS=*jrfile*]] [R=*guard*|*NONE|*SAME] [,W=*guard*|*NONE| *SAME] [,E=*guard*|*NONE|*SAME]

Die mit X markierten PLAM-Bibliothekselemente werden mit einem Guard-Schutz versehen. Falls keine Parameter angegeben sind, werden diese in einer gesonderten Maske angefordert.

%*name* Die Parameterangaben für die Variable Action ONXGUARD werden dem angegebenen Parameterset entnommen. Die Speicherung des Parametersets erfolgt mit dem CFS-Kommando SPGUARD %*name*.

JRS=*jrfile* Name der JRSAVE-Datei, in der der Parameterset gespeichert ist.
Standard: CFS.JRSAVE.*xxxx*. Näheres hierzu siehe Seite **565**.

R=*guard* Der lesende Zugriff auf das Datenobjekt wird mit einem Guard des angegebenen Namens geschützt. Ein Guard kann wie eine Datei mit PVS-ID, User-ID und Name angegeben werden. Der Name ist auf acht Stellen begrenzt.

R=*NONE Der lesende Zugriff wird weder durch Guard-, noch durch Basic-Access (BACL) Zugriffskontrollen überwacht.

R=*SAME Die Rechte des lesenden Zugriffs auf das Datenobjekt werden nicht verändert.

W=*guard* Der schreibende Zugriff auf das Datenobjekt wird mit einem Guard des angegebenen Namens geschützt.

W=*NONE Der schreibende Zugriff wird weder durch Guard-, noch durch Basic-Access (BACL) Zugriffskontrollen überwacht.

W=*SAME Die Rechte des schreibenden Zugriffs auf das Datenobjekt werden nicht verändert.

E=*guard* Der ausführende Zugriff auf das Datenobjekt wird mit einem Guard des angegebenen Namens geschützt.

E=*NONE Der ausführende Zugriff wird weder durch Guard-, noch durch Basic-Access (BACL) Zugriffskontrollen überwacht.

E=*SAME Die Rechte des ausführenden Zugriffs auf das Datenobjekt werden nicht verändert.

Hinweis:

Bei dem Schutz von PLAM-Elementen mit der Variablen Action ONXGUARD wird die Existenz der angegebenen Guards nicht geprüft. Wird eine Zugriffsart, z.B. R (Read) mit einem nicht existierenden Guard geschützt, so ist der entsprechende Zugriff (z.B. Read) für alle Benutzer solange gesperrt, bis ein Guard mit dem angegebenen Namen eingerichtet wird.

Zugriffsrechte (Basic Access Control List) für PLAM-Element ändern

Vorbemerkungen:

Für ein Element einer PLAM-Bibliothek gilt als User (Owner) die Benutzerkennung unter der die Bibliothek katalogisiert ist.

Die Einrichtung der BACL's steht ab BS2000 V10 zur Verfügung. Bei PLAM-Bibliotheken ist eine LMS-Version >= 2.0 erforderlich.

Bezüglich der Standard-Schutzattribute siehe Action-Code SP.

AM Access Modify.

Dieser Action-Code setzt voraus, daß bei der Selektion der Dateienliste die User Option BACL[F] angegeben wurde.
Die im Listeintrag dargestellten BACL-Attribute werden überschreibbar und können direkt modifiziert werden. Um eine Zugriffsart für eine bestimmte Benutzerdomäne zu verbieten, ist an der entsprechenden Stelle das Zeichen '-' einzutragen. Um einen Zugriff zu erlauben, ist die Bezeichnung der Domäne (UGO) bzw. die Art des Zugriffs (RWX) einzutragen.

AA[*x*] Access All.

x U | G | O:
Alle Zugriffsarten (d.h. Read/Write/Exec) werden für die angegebene Domäne erlaubt. Für *x* kann eine beliebige Kombination der Buchstaben U, G, O angegeben werden. Z.B. bewirkt der Action-Code AAUGO, daß alle Benutzer uneingeschränkten Zugriff auf das Element haben.

x W | R | X:
Allen Domänen (d.h. User/Group/Others) wird die angegebene Zugriffsart erlaubt. Für *x* kann eine beliebige Kombination der Buchstaben W, R, X angegeben werden.

Der Action-Code AA ohne einen Zusatz *x* bewirkt, daß das Element von allen Domänen ohne Einschränkung bearbeitet werden kann.

AN*x* Access No.

x U | G | O:
Alle Zugriffsarten (d.h. Read/Write/Exec) werden für die angegebene Domäne gesperrt. Der Eigentümer (User) kann diese Einstellung jederzeit wieder ändern. Für *x* kann eine beliebige Kombination der Buchstaben U, G, O angegeben werden.

x W | R | X:
Für alle Domänen (d.h. User/Group/Others) wird die angegebene Zugriffsart gesperrt. Für *x* kann auch eine beliebige Kombination der Buchstaben W, R, X angegeben werden.

AN No Basic Access Control List. Beim Zugriff auf das Element sind die BACL-Prüfungen nicht aktiv. Eine Standard Basic Access Control List kann z.B. durch Eingabe des Action-Codes AS wieder aktiviert werden.

AW*x* | **AR***x* | **AX***x*

Schreibenden/Lesenden/Ausführenden Zugriff für eine bestimmte Gruppe von Benutzerkennungen zulassen.

x Domäne, für die die angegebene Zugriffsart erlaubt werden soll.

- U User. Der Eigentümer (User-ID, unter der die Bibliothek katalogisiert ist) hat Zugriff auf das Element.
- G Group. Eine festgelegte Menge von Benutzerkennungen (Benutzergruppe) hat Zugriff auf das Element.
- O Others. Alle Kennungen, die nicht Eigentümer der Datei sind oder der Benutzergruppe des Eigentümers angehören, haben Zugriff auf das Element.
- N None. Weder User, noch Group, noch Others, d.h. niemand hat Zugriff auf das Element. Der Eigentümer (U) kann diese Einstellung jedoch jederzeit ändern.

Für *x* kann eine beliebige Kombination der Buchstaben U, G, O angegeben werden. Z.B. bewirkt der Action-Code AXUGO, daß Benutzer unter User, Group und Others ein EXEC auf das Element ausführen können.

AWN*x* | ARN*x* | AXN*x*

Verbot des schreibenden/lesenden/ausführenden Zugriffs für eine Gruppe von Benutzerkennungen.

x — Domäne, für die die angegebene Zugriffsart verboten werden soll. Beschreibung siehe oben. Für *x* kann auch eine Kombination der Buchstaben U, G, O angegeben werden. Z.B. bewirkt der Action-Code AWNGO, daß Benutzer aus der Gruppe und allen anderen Kennungen (Others) das Element nicht verändern dürfen.

AS — Access Rights = Standard. Es werden Standard BACL-Attribute für ein noch nicht BACL-geschütztes Element vergeben.

Hinweise:

Falls mit dem Action-Code A.. eine Basic Access Control Liste oder ein Passwort (siehe unten) für ein Element einer PLAM-Bibliothek definiert wurde, werden für diese Bibliothek auf Dateiebene die folgenden Prüfungen durchgeführt: Besitzt die Bibliothek das Attribut SHARE=YES ?. Wenn Ja: Ist für die Bibliothek eine Basic Access Control Liste (BACL) oder eine Full Access Control Liste (FACL) definiert ? Im negativen Falle werden für die Bibliothek die BACL-Attribute OWNER/GROUP/OTHERS = READ/WRITE/EXEC gesetzt. Damit ist sichergestellt, daß die Bibliothek mit dem Copy-Kommando nicht mehr kopiert werden kann.

Passwort zum Zugriff auf PLAM-Bibliothekselement definieren

Vorbemerkung: Die im folgenden beschriebene Möglichkeit, READ-/WRITE-/EXEC-Passworte für Elemente einer PLAM-Bibliothek zu definieren, setzt eine LMS-Version >= 2.0 und BS2000 >= 10.0 voraus.

ARP | AWP | AXP

Es wird eine Maske angefordert, in der der Benutzer ein bis zu vier Byte langes Read-, Write- oder Exec-Passwort für das betreffende Bibliothekselement vergeben kann. Das Eingabeformat für das Passwort ist [C]'...' bzw. X'...'.

Da die Passworte für Bibliothekselemente in verschlüsselter Form gespeichert werden, kann der Wert eines Passworts im Nachhinein weder vom Systemverwalter, noch vom Eigentümer der Bibliothek ermittelt oder abgefragt werden. Für nicht mehr bekannte Passworte bleibt für den Eigentümer der Bibliothek nur die Möglichkeit, diese zu entfernen und evtl. neu zu vergeben.

ARPN | AWPN | AXPN

Der Eigentümer, d.h. ein Benutzer, der unter der User-ID angeloggt ist, unter der die Bibliothek katalogisiert ist, kann ein vorhandenes Read-, Write- oder Exec-Passwort für das Bibliothekselement entfernen. Die Kenntnis des Passworts ist dabei nicht notwendig.

Action-Code C: PLAM-Bibliothekselement kopieren

C — Copy. Der Name des durch den Action-Code C markierten Datenobjekts wird in der Maske hell dargestellt und kann vom Benutzer überschrieben werden mit dem Namen, den die Kopie des Datenobjekts erhalten soll. Das ursprüngliche Datenobjekt wird dadurch nicht verändert. Der Namenseintrag in der Dateienliste wird nach der Copy-Operation wieder mit dem ursprünglichen Inhalt angezeigt.

Ein in Delta-Form gespeichertes Element einer PLAM-Bibliothek wird normalerweise in der Kopie als Voll-Element abgelegt. Ausnahme: Das Zielelement des Kopiervorgangs liegt bereits in delta-gespeicherter Form vor. Ein nicht in Delta-Form gespeichertes PLAM-Element wird auch in der Kopie als solches angelegt. Ausnahme: Das Zielelement des Kopiervorgangs liegt bereits in delta-gespeicherter Form vor.

CA Copy All. Bei einem deltagespeicherten Element einer PLAM-Bibliothek werden alle zugehörigen Versionen strukturerhaltend in ein Element mit dem angegebenen neuen Namen kopiert.

CS Copy ,SAME. Das PLAM-Bibliothekselement wird mit den gleichen Schutzattributen kopiert. Diese sind z.B. Read-, Write- und Exec-Passworte sowie die Basic Access Control Liste (BACL). Standardmäßig werden Elemente ohne Übernahme der Schutzattribute kopiert.

Action-Code E: PLAM-Bibliothekselement löschen

E Bibliothekselement löschen.

EN Das Bibliothekselement wird gelöscht, wobei die ERT-Option (Erase with retain Tempfile) für diesen Löschvorgang explizit ausgeschaltet ist.

Im Zusammenhang mit dem Löschen von Datenobjekten wird auf den in CFS standardmäßig eingeschalteten ERT-Modus hingewiesen.
Der ERT-Modus (Erase with Retain Tempfiles) hat zur Folge, daß die mit dem Action-Code E gelöschten Bibliothekselemente zunächst in entsprechende Elemente in eine als Temporärdatei angelegte Bibliothek übertragen werden. Diese temporären Bibliotheken werden vom BS2000 bei Prozeßende gelöscht. Versehentlich vom Benutzer mit dem Action-Code E markierte und gelöschte Daten sind damit noch bis Prozeßende physisch vorhanden (zugreifbar durch FILENAME-SELECT : LMS #*bibl*).

Der Name der temporären Bibliothek wird gebildet, indem dem Namen der Bibliothek, aus der Elemente gelöscht werden, das Tempfile-Zeichen # vorangestellt wird.

EA Bei delta-gespeicherten Elementen einer PLAM-Bibliothek bewirkt der Action-Code E das Löschen der angezeigten Version. Alle übrigen Versionen des Elements werden nicht gelöscht. Um alle Versionen eines Elements zu löschen, ist der Action-Code EA (Erase All) zu verwenden.

Action-Code EDT: PLAM-Bibliothekselement im EDT bearbeiten

EDT[*n*] | EDF[*n*] Das Bibliothekselement kann im Formatmodus des EDT bearbeitet werden.
Rückkehr in die CFS-Maske durch die K1-Taste oder das EDT-Kommando H [ALT].

UPD[*n*] Bibliothekselement aus dem virtuellen EDT-Bereich *n* zurückschreiben.

Der Action-Code ist nach der Rückkehr aus dem EDT bereits im Action-Feld vorgegeben. Falls das Datenobjekt nicht zurückgeschrieben werden soll, ist der Action-Code mit Blanks zu überschreiben.

Die Quittung der einzelnen Updates erfolgt durch U---U am rechten Ende der Bildschirmzeile.

Ab LMS Version 1.4 können mit den Action-Codes EDT/UPD auch in Delta-Form gespeicherte Elemente einer PLAM-Bibliothek bearbeitet und zurückgeschrieben werden. Bei UPD wird dabei jeweils ein neues Element mit einer um 1 erhöhten Versionsnummer erzeugt.

Administrationsrechte für eine PLAM-Bibliothek anzeigen / ändern

Vorbemerkungen:

Die i.f. beschriebene Möglichkeit der Vergabe von Administrationsrechten für eine PLAM-Bibliothek, setzt eine LMS-Version >= 2.0 und BS2000 >= 11.0 voraus.

Der Action-Code LA [*xx*] kann in der Dateienliste bei PLAM-Bibliotheken (FCBTYP = PL) und bei beliebigen Elementen einer PLAM-Bibliothek angegeben werden. Die Wirkung ist in jedem Fall die gleiche: Es werden die Administrationsrechte der aktuellen Bibliothek angezeigt bzw. verändert. Befindet sich der Benutzer innerhalb einer Bibliothek, so ist das Element, bei dem der Action-Code angegeben wird, ohne Bedeutung.

LA

Es werden die Administrationsrechte, die Standard Elementschutzattribute sowie die Standard Schutzattribute für Elementtypen für die aktuelle Bibliothek angezeigt. Die Informationen werden in folgendem Format dargestellt:

```
%  SHOW LIBRARY ADMINISTRATION + STANDARD ELEMENT-PROTECTION
%
%  LIBRARY NAME           : CFSLIB
%
%  LAST ACCESS DATE       : YES              (Modify with:  LAKL/LANKL)
%
%  LIBRARY ADMINISTRATION : OWNER  = YES    GROUP  = NO      OTHERS  = NO
%                           PASSW  = NO
%
%  STD ELEMENT-PROTECTION : OWNER  = R W *  GROUP  = - - *  OTHERS  = - - *
%                           RDPASS = NO     WRPASS = NO      EXPASS  = NO
%
%  STD PROTECTION TYPE C/ : OWNER  = R W X  GROUP  = - - X  OTHERS  = - - X
%                           RDPASS = NO     WRPASS = NO      EXPASS  = NO
%                           ADMIN  = OWNER           OTHERS
%                           WRITE-CONTROL = YES
%
%  PLEASE ACKNOWLEDGE
```

Hinweise:

Das Zeichen '*' in der Protection-Zeile zeigt an, daß für die entsprechende Zugriffsart keine Basic Access Control Liste definiert ist. Dies bedeutet, daß Owner, Group und Others Zugriff in der entsprechenden Art auf das Element haben.

Die Definition von Standard Element- und Elementtyp-Schutzattributen ist mit dem Action-Code SP möglich. Näheres hierzu auf Seite **204**.

LA *x* [*t*/]

Administration der gesamten Bibliothek bzw. der Elemente eines Typs bestimmten Benutzern erlauben. Das Administrationsrecht gestattet es, Elemente neu aufzunehmen, zu löschen, zu kopieren oder umzubenennen. Das Administrationsrecht kann dem Eigentümer (USER, d.h. der Benutzerkennung, unter der die Bibliothek katalogisiert ist), der Benutzergruppe, der der Eigentümer angehört (GROUP) oder allen anderen Benutzerkennungen (OTHERS) zugeteilt werden.

x — Domäne, der die Administration der Bibliothek erlaubt werden soll.

- U — User. Der Eigentümer (alle Benutzer der Kennung, unter der die Bibliothek katalogisiert ist) hat das Recht zur Administration.
- G — Group. Eine festgelegte Menge von Benutzerkennungen (Benutzergruppe) soll das Administrationsrecht besitzen.
- O — Others. Alle Kennungen, die nicht Eigentümer der Bibliothek sind oder der Benutzergruppe des Eigentümers angehören, sollen das Administrationsrecht besitzen.

Für *x* kann eine beliebige Kombination der Buchstaben U, G, O angegeben werden. Z.B. bewirkt der Action-Code LAUG, daß die Benutzerkennung, unter der die Bibliothek katalogisiert ist und die Benutzergruppe das Administrationsrecht für die Bibliothek besitzen.

t/ — Die zuvor angegebene Domäne (U/G/O) soll das Administrationsrecht für Elemente des Typs *t* erhalten. Nur Benutzer dieser Domäne haben das Recht, Elemente des Typs *t* zu ändern, zu löschen oder umzubenennen.
Durch wiederholtes Eintragen des Action-Codes kann das Administrationsrecht mehreren Domänen zugeteilt werden. Beispiel: LAUS/ LAOS/ erlaubt dem Owner (User) und der Domäne Others die Administration der Elemente des Typs S.
Das typbezogene Administrationsrecht für Elemente steht erst ab LMS Version 3.0 zur Verfügung.

LAN *t*/ — Durch den Action-Code LAN*t*/ wird eine zuvor definierte Administrationsberechtigung für Elemente des Typs *t* zurückgesetzt. Für diesen Typ gelten dann die globalen Administrationsrechte.

Der Action-Code LA kann bei einem beliebigen Element der Bibliothek eingetragen werden. Die Wirkung ist in allen Fällen die gleiche.

Im Standardfall, d.h. ohne besondere Definition eines Administrationsrechts für einen Elementtyp gelten die globalen Administrationsrechte der Bibliothek.

LAN *x* — Recht der Administration der Bibliothek für Benutzerkennungen verbieten.
Ohne Administrationsrecht kann der Benutzer Elemente nicht neu aufnehmen, löschen, kopieren oder umbenennen.

x — Domäne, der die Administration der Bibliothek nicht erlaubt werden soll. Beschreibung siehe oben. Für *x* kann auch eine Kombination der Buchstaben U, G, O angegeben werden.

LAN — Administrationsrechte für die Bibliothek werden zurückgesetzt. Jeder Benutzer, der gemäß dem DVS Zugriff auf die Bibliothek hat, kann Elemente aufnehmen, löschen, kopieren oder umbenennen.

LAP *t*/ — Es wird eine Maske angefordert, in der der Benutzer ein bis zu vier Byte langes Administrations-Passwort vergeben kann. Administrationstätigkeiten in der Bibliothek (siehe oben) sind dann nur möglich, falls zuvor durch ein Password-Kommando das benötigte Password abgesetzt wurde. Die Möglichkeit der Vergabe eines Administrationspasswortes ist unabhängig von der Definition bestimmter Administrationsdomänen mit dem Action-Code LA *x* (siehe oben).
Die Eingabe des Administrations-Passworts erfolgt in der Form: [C]'...' oder X'...'.

Da Passworte in PLAM-Bibliotheken in verschlüsselter Form gespeichert werden, kann der Wert eines Passworts im Nachhinein weder vom Systemverwalter, noch vom Eigentümer der Bibliothek ermittelt oder abgefragt werden. Für ein nicht mehr be-

kanntes Passwort bleibt für den Eigentümer der Bibliothek nur die Möglichkeit, dieses zu entfernen und evtl. neu zu vergeben.

LANP Der Eigentümer, d.h. ein Benutzer, der unter der User-ID angeloggt ist, unter der die Bibliothek katalogisiert ist, kann ein vorhandenes Administrations-Passwort entfernen. Es ist hierzu nicht die Kenntnis des betreffenden Passworts notwendig.

LAKL | LANKL Keep Last Access. Das Aktualisieren des Last Access Date für Zugriffe auf Elemente der Bibliothek wird ein-/ausgeschaltet. Die Variante KL des Action-Codes LA ist ab LMS Version 3.0 zulässig. Im Modul CFSMAIN kann der Systemverwalter eine Option aktivieren, die bewirkt, daß von CFS neu angelegte Bibliotheken automatisch mit Last Access Date eingerichtet werden.

LAWC | LANWC Die Eigenschaft Write-Control für alle Elemente mit dem gleichen Typ wie das mit dem Action-Code LAWC markierte Element ein-/ausschalten.
Die Variante WC des Action-Codes LA ist ab LMS Version 3.0 zulässig.

Bei eingeschaltetem Write-Control für einen Elementtyp *t* gelten folgende Zugriffsbeschränkungen für Elemente des Typs *t*:

- Falls ein Element *t/element/version* nicht ausgeliehen, d.h. im Zustand HOLD ist, ist es nicht möglich, dieses Element zu ändern oder zu überschreiben.
- Unabhängig vom Hold-Status ist es nicht möglich, ein Element *t/element/ version* umzubenennen.
- Das Element *t/element/version* darf nicht ausgeliehen sein (Hold-Status), damit aus einer Version eine höhere Version erzeugt werden kann.

Hinweis:

Falls mit dem Action-Code LA.. das Administrationsrecht für die PLAM-Bibliothek verändert wurde, werden von CFS für diese Bibliothek auf Dateiebene die folgenden zusätzlichen Prüfungen durchgeführt: Besitzt die Datei das Attribut SHARE=YES ?. Wenn Ja: Ist für die Datei eine Basic Access Control Liste (BACL) oder eine Full Access Control Liste (FACL) definiert ? Im negativen Falle werden für die Datei die BACL-Attribute OWNER/GROUP/OTHERS = READ/WRITE/EXEC gesetzt. Damit ist sichergestellt, daß die Datei nicht mehr kopiert werden kann.

Guard-Schutz für Bibliothekselemente festlegen

SG Set Guards. PLAM-Bibliothekselement mit Guards schützen. Falls die Guard-Profile (Read-/Write-/Exec-Guard) dem CFS noch nicht bekannt gemacht wurden - entweder durch ein früheres GUARD-Kommando/Variable Action oder durch einen früheren Action-Code SG/SGM -, werden diese Angaben beim ersten SG Action-Code in einer Menue-Maske angefordert. Für eine Beschreibung der Menue-Maske siehe Variable Action ONXGUARD, Seite 147.

SGM wie Action-Code SG, jedoch wird in jedem Fall die Guards-Maske ausgegeben.

SGN Set Guards None. Der Guard-Schutz wird für das betreffende Datenobjekt außer Kraft gesetzt. Bei Dateien kann dies auch durch das BS2000-Kommando /MODIFY-FILE-ATTR *datei*,PROTECTION(GUARD=*NONE) erreicht werden.

Standard-Schutzattribute für neue Elemente einer PLAM-Bibliothek festlegen

Vorbemerkung:

Die im folgenden beschriebene Möglichkeit der Vergabe von Standard-Schutzattributen für neue Elemente einer PLAM-Bibliothek, setzt eine LMS-Version >= 2.0 und BS2000 >= 10.0 voraus.

SPSTD Set Protection Standard. Die Schutzattribute (BACL's) des mit dem Action-Code SP bezeichneten Bibliothekselements werden als Standard-Schutzattribute für alle neu erzeugten Elemente in der PLAM-Bibliothek verwendet. Sind für das bezeichnete Bibliothekselement Write-, Read-, Exec-Passworte definiert, so werden diese nicht in die Standard-Schutzattribute übernommen.

Die Standard-Schutzattribute gelten nicht für neue Elemente eines Elementtyps *t*/, für den ein eigener Standardschutz definiert wurde (siehe unten, Action-Code SP*t*/). Die Standard-Schutzattribute gelten ebenfalls nicht für Elemente, die mit dem Action-Code CA (Copy Same) kopiert, bzw. mit der Variablen Action ONXADD.. ,CPR (Copy Protection) aus einer anderen Bibliothek übertragen wurden.

Die aktuell gültigen Standard-Schutzattribute werden durch den Action-Code LA (Library Attributes, siehe Seite 197) angezeigt.

SP *t*/ Set Protection for Type *t*. Die Schutzattribute (BACL's) des mit dem Action-Code SP bezeichneten Bibliothekselements werden als Standard-Schutzattribute für alle in die PLAM-Bibliothek neu aufgenommenen Elemente des Typs *t* verwendet. Sind für das bezeichnete Bibliothekselement Write-, Read-, Exec-Passworte definiert, so werden diese nicht in die Standard-Schutzattribute für den Elementtyp übernommen.

SPN *t*/ Die Standard-Schutzattribute für neue Elemente des angegebenen Typs werden zurückgesetzt. Die typunabhängigen Standard-Schutzattribute (siehe Action-Code SPSTD) bleiben nach wie vor erhalten.

SPN Die Standard-Schutzattribute für neue Elemente in der Bibliothek werden zurückgesetzt. Die Standard-Schutzattribute für für bestimmte Elementtypen bleiben dabei nach wie vor bestehen.

Hinweis:

Falls mit dem Action-Code SP.. Standard-Schutzattribute für die PLAM-Bibliothek definiert oder verändert wurden, werden von CFS für diese Bibliothek auf Dateiebene die folgenden zusätzlichen Prüfungen durchgeführt: Besitzt die Datei das Attribut SHARE=YES ?. Wenn Ja: Ist für die Datei eine Basic Access Control Liste (BACL) oder eine Full Access Control Liste (FACL) definiert ? Im negativen Falle werden für die Datei die BACL-Attribute OWNER/GROUP/OTHERS = READ/WRITE/ EXEC gesetzt. Damit ist sichergestellt, daß die Datei mit dem Copy-Kommando nicht mehr kopiert werden kann.

17. CFS-Benutzerexits

Unter CFS können externe Routinen aufgerufen werden als:

a) User Options zur Auswertung der Katalogeinträge nach speziellen Merkmalen.

b) Variable Actions zur Realisierung besonderer Verarbeitungsarten für Dateien /Bibliothekselemente/Jobvariablen.

c) Action-Codes zur Realisierung besonderer Verarbeitungsarten für Dateien/ Bibliothekselemente/Jobvariablen über eigene Action-Codes.

d) Kommandos zur Realisierung besonderer Verarbeitungsarten für Dateien/ Bibliothekselemente/Jobvariablen.

Der CFS-Anwender kann eine externe User Option bzw. Kommandoroutine in folgender Form aufrufen:

(*modul* [, *bibl*]) [*params*]

Der Aufruf einer externen Routine für eine Variable Action erfolgt in der Form:

ONX(*modul* [, *bibl*]) [*params*] | ON&(*modul* [, *bibl*]) [*params*]

modul Name der aufzurufenden Routine (Modul-/CSECT-Name).

bibl Bibliothek, aus der die Benutzerroutine nachzuladen ist.
Standard: CFSLIB

params wahlweise Parameter, die der Benutzerroutine beim Aufruf mitgegeben werden und die deren Ausführung steuern.

Der Aufruf einer externen Action-Code Routine geschieht durch Angabe des betreffenden Action-Codes in der Action-Spalte.
Damit CFS diesen Action-Code richtig verarbeitet, muß er vom Systemverwalter in eine CFS-interne Action-Code Tabelle eingetragen werden. Genaueres hierzu finden Sie im Quellprogramm CFSMAIN in der CFS.S.LMSLIB.

Aufrufkonventionen für externe Routinen:

R1 Adresse der Parameterliste
R13 Adresse einer 72 Byte langen Save-Area
R14 Returnadresse in CFS
R15 Ansprungadresse der Benutzerroutine

Die **Registersicherstellung** sollte nach den Standardkonventionen erfolgen:

```
STM     14,12,12(13)     (Ansprung)
. . .
LM      14,12,12(13)     (Rückkehr)
```

In der ausgelieferten LMS-Bibliothek CFS.S.LMSLIB befinden sich Quellprogramme als Beispiele zu den einzelnen Arten von externen Routinen:

a) **FREE** als Beispiel einer Routine zur Realisierung eine benutzereigenen User Option (Free Space).

b) **CONV** als Beispiel einer Routine für eine benutzereigene Variable Action.

c) **PREMOTE** als Beispiel einer Routine für einen benutzereigenen Action-Code (PD).

d) **VSNSP** als Beispiel einer Routine für ein benutzereigenes CFS-Kommando.
VSNSP gibt für die spezifizierte Datei (siehe unten) alle VSNs, auf denen die Datei Speicherplatz belegt hat, sowie für jede VSN die Anzahl der auf ihr belegten PAM-Seiten. Die Aufrufsyntax lautet: (VSNSP) *datei*.

Hinweise:

Die beim Aufruf der externen Routinen benutzten Eingabeparameterleisten sind in den oben erwähnten Beispielprogrammen intern dokumentiert.

Benutzerdefinierte Action-Codes müssen, um als solche erkannt zu werden, dem CFS-System bekannt gemacht werden. Hierzu existiert eine besondere Action-Code Tabelle. Die Struktur des Tabellenelements für den Action-Code PD ist im Quellprogramm CFSMAIN in der CFS.S.LMSLIB beschrieben.

Auch im Connection-Handler besteht die Möglichkeit einer eigenen Nachrichtenbehandlung. Für nähere Informationen hierzu kontaktieren Sie bitte die OPG Online-Programmierung.

CFS als Unterprogramm

CFS kann von beliebigen Benutzerprogrammen aufgerufen werden, ohne daß diese dabei entladen werden. Das Unterprogramm CFS kann direkt in die Benutzerprogramme eingebunden oder bei Bedarf durch einen Link-Makro nachgeladen werden. Zum Nachladen von CFS ist folgender Link-Makro abzusetzen:

```
LINK ENTRY=CFSMAIN,LIBNAM=$TSOS.CFSLIB,INHIBIT=YES
```

Beim Laden von CFSMAIN werden weitere Module per Autolink-Funktion nachgezogen. Daher bietet es sich an, gleich einen Großmodul über LINK-Makro zu laden, der die notwendigen Untermodule CFSUP und GDATE bereits enthält. Dieser Großmodul CFS2 kann durch /DO CFS. S.LMSLIB(CFSLNK2) erzeugt werden.

Der Modul CFSMAIN bzw. der Großmodul CFS2 muß auf **Seitengrenze** im unteren Adressraum geladen werden.

Der Modul CFSMAIN liegt in der Bibliothek CFS.S.LMSLIB im Sourcecode vor und kann zur Initialisierung von RZ- bzw. benutzerspezifischen Standardwerten angepaßt werden.

CFS aktiviert intern Stxit's für INTR-, ABEND-, K2- (ESCPBRK) und SVC-Ereignisse.

Das Hauptprogramm kann CFS Selektionsparameter, Kommandos, Action-Codes, sowie Daten zur Modifikation von Dateien mitgeben. Damit besteht die Möglichkeit, beim Aufruf von CFS durch ein übergeordnetes Hauptprogramm bereits eine Reihe von Aktionen ausführen zu lassen, z.B. Dateien / Elemente aus einer bestimmten Bibliothek selektieren, Eröffnen von Connections, Laden von P-Tasten, bestimmte CFS-Modi einstellen, bestimmte Dateien/Bibliothekselemente selektieren, Dateien in den EDT laden, Dateien im Display von CFS anzeigen usw.

Die verschiedenen Nutzungsmöglichkeiten von CFS als Unterprogramm werden am besten durch die Beschreibung der Aufrufparameterliste erläutert.

Aufrufkonventionen und Parameterliste: Hauptprogramm --> CFS

Registerbelegung

R1	Adresse der Parameterliste für CFSUP.
R13	Adresse einer 18 Worte langen Save-Area im aufrufenden Programm.
R14	Rückkehradresse in das Hauptprogramm.
R15	Einsprungadresse: Beginn des Moduls CFS2 bzw. CFSMAIN.

Parameterliste

```
P1    DC    A(RECTABLE) | A(0)
P2    DC    A(IND)
      DC    A(3)                  optional, nur bei IND = C'N' sinnvoll
P3    DC    A(CFSLIB)             optional, nur bei IND = C'N' sinnvoll
```

```
P1    DC    A(RECTABLE)
```

Liste von Eingabesätzen im RDATA-Format.

CFS verhält sich wie in einer Dialogprozedur (siehe Seite 489). Insbesondere wird die durch einen *002-Satz definierte Selektion ausgeführt.

RECTABLE Liste von Eingabesätzen im V-Format. RECTABLE hat die gleiche Funktion wie SYSDTA, nur daß die Datensätze im virtuellen Speicher und nicht in einer Datei übergeben werden. Das Ende von RECTABLE ist durch das Zeichen X'80' zu markieren. Nach Erreichen des Endes von RECTABLE, bzw. nach Lesen eines *DIA-Satzes (siehe Seite 499) erwartet CFS alle weiteren Eingaben vom Bildschirm.

```
P1    DC    A(0)
```

CFS erhält vom aufrufenden Programm keine Eingaben.

Nach dem Ansprung durch das Benutzerprogramm kommt CFS am Bildschirm mit der Selektionsmaske wie nach einem normalen EXEC durch den Benutzer. Die Rückkehr in das Hauptprogramm kann erfolgen durch die Kommandos END / * / MAIN.

```
P2    DC    A(IND)
```

```
IND      DC    C'N'
```

New CFS beginnt mit einem neuen Verarbeitungsgang, d.h. es wird als erstes ein Satz zur Spezifizierung der Selektionsbedingungen erwartet (*001-, bzw. *002-Satz).

```
IND      DC    C'C'
```

Continue Fortsetzungsaufruf. CFS fährt an der Stelle fort, an welcher es beim letzten Aufruf aufgrund der Kommandos END / * / MAIN unterbrochen wurde.

```
P3    DC    A(CFSLIB)
```

```
CFSUID   DC    CL54'$userid.CFSLIB'
```

Name der eingesetzten CFSLIB.
Falls angegeben, wird aus dieser Angabe die Kennung, unter der CFS gespeichert ist ermittelt. Die Kennung wird anstelle der Standard-kennung $TSOS. vor die Namen der internen CFS-Dateien CFSLIB, CFSHELP, CFSCOM, CFS.MESSAGE, CFSPROT, CFSTERM, CFS.PDFILE, CFS.INF.FORMAT und CFSUSER gesetzt.

Beispiel:

In der CFS.S.LMSLIB finden Sie unter dem Namen TEST ein einfaches Musterprogramm, in dem CFS mehrere Male als Unterprogramm aufgerufen wird. Das Programm kann mit /DO CFS.S.LMSLIB(ASSEMB),TEST übersetzt und mit /EXEC (TEST, CFSLIB) aufgerufen werden.

Im folgenden Beispiel ist eine Parameterleiste des Hauptprogramms für CFS dargestellt.

```
P1       DC    A(RECTABLE)
P2       DC    A(IND)
IND      DC    C'N'
```

```
RECTABLE DS    0A
REC1     DC    AL2(REC1L,0)
         DC    C'*002 FILE=NO'
REC1L    EQU   *-REC1
REC2     DC    AL2(REC2L,0)
         DC    C'OC1/LOGON user-id,acct,C''passw'''
REC2L    EQU   *-REC2
REC3     DC    AL2(REC3L,0)
         DC    C'LOAD programm'
REC3L    EQU   *-REC3
REC4     DC    AL2(REC4L,0)
         DC    C'^'
REC4L    EQU   *-REC4
REC5     DC    AL2(REC5L,0)
         DC    C'PL plus-modul,maplib'
REC5L    EQU   *-REC5
REC6     DC    AL2(REC6L,0)
         DC    C'EDT,datei'
REC6L    EQU   *-REC6
REC7     DC    AL2(REC7L,0)
         DC    C'@RET'
REC7L    EQU   *-REC7
REC8     DC    AL2(REC8L,0)
         DC    C'NP'
REC8L    EQU   *-REC8
         DC    X'80'
               |
               |
               ▫---- Es folgt kein weiterer Datensatz
```

Hinweis:

Bei Übergabe einer RECTABLE sollten vor dem Ansprung von CFS die Prozeßschalter 1 und 4 gesetzt werden (SETSW ON=(1,4)).

18. Parameter ändern

Allgemeine Bemerkungen zum Setzen und Rücksetzen von Parametern

Die im folgenden beschriebenen Kommandos - einzugeben im Feld COMMAND in der Maske der Dateienliste - dienen dazu, gewisse CFS-Funktionen ein- bzw. auszuschalten. Die entsprechenden Kommandos sind dabei fast immer nach folgender Regel konstruiert: Das Ausschalten eines CFS-Verarbeitungsmodus erfolgt durch Voranstellen des Buchstabens N (No) vor das entsprechende Einschaltkommando.

Beispiele:

HC	Hardcopy	<-->	NHC	No Hardcopy
DL	Display Long	<-->	NDL	No Display Long
H	Hexadezimal Mode	<-->	NH	No Hexadezimal Mode

Für einige Kommandos wurde auch eine Angleichung an entsprechende EDT-Kommandos vorgenommen. Es ergeben sich dadurch oft mehrere Kommandos für ein und dieselbe Funktion:

EL	Edit Long	<-->	ELO	Edit Long Off
LOW	LOWer On	<-->	LOWO	LOWer Off
SC	SCale On	<-->	SCO	SCale Off

SET-Kommando: CFS-Parameter festlegen

SET Es wird eine Maske ausgegeben, in der die Belegungen der am häufigsten benutzten Parameter angezeigt und für den aktuellen Lauf verändert werden können.

SET [%*name* | %? [,*jrsave*]]

Die Inhalte der in der SET-Maske aufgeführten Parameter werden mit den in dem Parameterset gespeicherten Angaben modifiziert. Die Ausgabe der SET-Maske wird unterdrückt.

Das Abspeichern der zuletzt verwendeten SET-Parameter in einen Parameterset erfolgt mit dem Kommando SP SET%*name*. Näheres hierzu auf Seite 283.

Die aufgrund des SET-Kommandos ausgegebene Maske hat folgendes Format:

```
                     C F S   Parameter-Settings

User-Library                           (UL): CFS.USERLIB
Standard User-Option(s)                (UO):
EDT Initialisation-commands       (EDTINIT):
Standard User-ID                      (UID):
Standard Sort-Option                   (SO): NAME,A      Hardcopy     (HC/NHC): OFF
Query on  Copy: OFF   Add: OFF    Select: ON   Erase: OFF   Term: OFF  EDT: ON
Show used resources  (M/CPU/WSI/IO/OFF): OFF  Display          (AGE/DATE): AGE
Erase with Retain Tempfiles  (ERT/NERT): ON   Display        (LASTP/ALLOC): LASTP
Duplicate Key mode          (DUPK/NDUPK): OFF  PLAM :  Age as   (SDAT/UDAT): SDAT
Command-characters for Isam-Editing (RD)       Copy : C    Elim : E   Retain: R
           Insert: I    Move : M    Join : J    Clear: *    After: A   Before: B
Transform lowercase letters    (LOW/CAP): CAP CFS Command-Separator   (SEP): ;
Check PLAM-Types on Add     (CT/CTQ/NCT): CTQ INF Standard-Format (I FORM):

                             Connection: 0 1 2 3 4 5 6 7 8 9
Direct K2-mode ON/OFF (1/0) (DKCn/NDKC): 0 0 0 0 0 0 0 0 0 0
Asynchron Mode                (ACn/NACn): 0 0 0 0 0 0 0 0 0 0
Hardcopy in Connections       (HCn/NHCn): 0 0 0 0 0 0 0 0 0 0
Key for single Mask HC in Conn.     (CH): NONE Show Conn. Line     (SCL/NSCL): OFF
Automatic Connection Display (ACD/NACD): ON   DO-Params in Conn.    (DP/NDP): ON
Symbol to invoke Command-Memory         : -   Escape-Symbol in  Connections: ^
BS2000-Prompt in Connections        (CP): /
```

Hinweise:

Alle Parameter können auch mit den in der SET-Maske aufgeführten Einzelkommandos (xxx/yyy) gesetzt oder zurückgesetzt werden. Falls die Einzelkommandos in der SET-Maske nicht aufgeführt sind, so sind die Einzelkommandos gleichlautend mit den in Großbuchstaben angegebenen Alternativen, z.B. "Display AGE/DATE". In diesem Fall lauten die Einzelkommandos AGE/DATE. Die Einzelkommandos zum Umdefinieren der Zeichen für das Kommandogedächtnis und das Connection-Fluchtsymbol sind auf Seite 374 beschrieben.

Nicht für alle in CFS einstellbaren Parameter existiert in der SET-Maske ein entsprechendes Feld bzw. ein Einzelkommando. Diese sind im allgemeinen RZ-globale Parameter und können nur über das Modify-Kommando von CFS (M L'...',C'...') benutzerindividuell angepaßt werden. Mehr hierzu siehe Seite 479.

Das Setzen von benutzerspezifischen Parametern kann automatisiert erfolgen durch Eintragen der Einzelkommandos in eine CFS-Startup-Datei.

Beispiel für das Setzen von Parametern mit einer Startup-Datei:

`*002 FILE=NO`	Verzweigen zum Kommandofeld der zweiten CFS-Maske
`UO EXTNT`	Default User Option definieren
`ALLOC`	Anzeige von Allocated Space anstelle Last-Page
`SO AGE,A`	Sort Option: Dateienliste aufsteigend nach AGE sortieren
`DUM`	Display Used Memory anstelle der Uhrzeit
`CP '&CON /'`	Connection Prompt: n / (n = Conn.-Nr)
`CRD -=§`	Zeichen für Kommandogedächtnis in Connections ändern
`NP`	Verzweigen in die CFS-Selektionsmaske

Für eine ausführliche Darstellung der Möglichkeiten des Startup-Mechanismus von CFS wird auf Seite **537** und folgende (Startup) verwiesen.

Add Compressed (LMS-ISAM)

AC | NAC — Add Compressed/No Add Compressed.

AC — Komprimierte Aufnahme von Dateien in LMS-ISAM Bibliotheken.

NAC — Entkomprimierte Aufnahme von Dateien in LMS-ISAM Bibliotheken.
Standard: AC (komprimierte Aufnahme).

Einfügeposition für kopierte Dateien

SET COPYMODE=A|E Einfügeposition der durch den Action-Code C neu erzeugten Kopien in der Dateienliste:

A — direkt hinter der Ursprungsdatei

E — am Ende der Dateienliste.

Standard: SET COPYMODE=E.

Behandlung von Zeichenfolgen im Suche-Kommando

SET SEARCHMODE=A|C

A — Im Suche-Kommando wird '...' bzw. C'...' wie A'...' behandelt.

C — Im Suche-Kommando wird '...' bzw. C'...' nicht wie A'...' behandelt.

Standard: SET SEARCHMODE=C.

Verhalten beim Suchen mit Ersetzen von Zeichenfolgen

SET REPLACEMODE=L|F

L — Bei einem Suche-Kommando der Art S,'123'*'890'='abc' wird bei Treffern die Zeichenfolge '890' durch 'abc' ersetzt.

F — Bei einem Suche-Kommando der Art S,'123'*'890'='abc' wird bei Treffern die Zeichenfolge '123' durch 'abc' ersetzt.

Standard: SET REPLACEMODE=F.

Automatischer / nicht automatischer Connection-Display

<u>ACD</u> | NACD   Automatic Connection Display / No Automatic Connection Display.

ACD   Bei jeder Betätigung der **K2**-Taste erfolgt die Anzeige aller im Moment eröffneten Connections.

NACD   Der automatische Connection Display bei **K2** wird nicht ausgeführt.

Standard:	ACD	bei SETSW OFF=(4)
	NACD	bei SETSW ON=(4)

Automatic Documentation

<u>ADOC</u> | NADOC   Automatic Documentation File / No Automatic Documentation File

ADOC   Bei jeder neuen Zusammenstellung einer Dateienliste (Selektionsmaske/NP-Kommando) wird die alte Dateienliste in eine temporäre Datei gesichert.
Standard: ADOC.

Hinweise:

Der Name der Zwischendatei lautet #CFS.SAVELIST.*jobname* bzw. #CFS.SAVE LIST.*acct-nr* (*acct-nr* nur, falls im LOGON-Kommando kein Jobname angegeben wurde). Das standardmäßige Prefix #CFS.SAVELIST. kann vom CFS-Administrator auch modifiziert werden.

Durch Eingabe von RL (Restore List) im Feld FILENAME-SELECT wird die letzte Dateienliste wieder angezeigt.

Mit dem Kommando RL kann die Anzeige zwischen der letzten und der aktuellen Dateien-/Bibliotheksliste alterniert werden. Das Kommando RL erlaubt somit einen raschen Wechsel zwischen zwei verschiedenen Dateien-/Bibliothekslisten.

Dateienliste: Anzeige der letzten Veränderung als Anzahl von Tagen

AGE   Altersangaben, insbesondere die Angaben in der AGE-Spalte der Dateienliste, werden als Anzahl von Tagen dargestellt. Durch das Kommando DATE kann das Alter auch in Form einer Datumsangabe angezeigt werden. Anstelle von AGE kann auch das Kommando NDATE angegeben werden.
Standard: AGE.

Versionshochzählung bei ADD auf bestehende Elemente in LMS-/PLAM-Bibliotheken

AI | <u>NAI</u>   Add with Increment of version/No Add with Increment of version.

AI   Automatische Versionserhöhung bei ADD in LMS-Bibliotheken (LMS-ISAM/ PLAM): Falls ein Element gleichen Namens in der Bibliothek bereits existiert, wird die Versionsbezeichnung des neu aufzunehmenden Elements um 1 erhöht.

NAI Beim ADD in LMS-Bibliotheken wird, falls bereits ein Element gleichen Namens in der Bibliothek existiert, dasjenige mit der höchsten Versionsnummer überschrieben. Dies gilt jedoch nicht, falls es sich um ein in Delta-Form gespeichertes Element in einer PLAM-Bibliothek handelt. In diesem Fall wird auch im NAI-Modus eine automatische Erhöhung der Versionsnummer um 1 vorgenommen.
Standard: NAI.

Allocated Space als Merkmal für Dateigröße

ALLOC In der ersten Spalte der Dateienliste wird als Maß für die Größe einer Datei der Wert für Allocated Space (zugewiesene PAM-Seiten) angezeigt.

Standardmäßig wird für die Dateigröße die Anzahl der beschriebenen PAM-Seiten (LASTP) angezeigt.

Kommandogedächtnis automatisch komprimieren

AMC | NAMC Automatic Memory Compress/No Automatic Memory Compress.

AMC Jedes der CFS-internen Gedächtnisse (FILENAME-SELECT, COMMAND, Connection 0, ..., 9) besteht aus einer Speichertabelle von 4096 Bytes, in die die Tastatureingaben der Reihe nach eingetragen werden. Bei Erreichen des Endes einer dieser Tabellen werden die Tabelleneinträge am Anfang zyklisch überschrieben. Um dieses Überschreiben der im Kommandogedächtnis festgehaltenen früheren Eingaben zu verzögern und noch mehr Freiplatz zu gewinnen, wird intern ein Memory Compress durchgeführt. Der Memory-Compress bewirkt, daß gleiche Eingaben aus dem Gedächtnis eliminiert werden und dadurch mehr freier Platz in der Tabelle zur Verfügung steht. Nach der Komprimierung entspricht die Reihenfolge der Eingaben im Kommandogedächtnis nicht mehr der tatsächlichen Reihenfolge der getätigten Eingaben.
Standard: AMC.

Zum Thema Kommandogedächtnis siehe auch CM (Compress Memory, Seite 223), SM (Save Memory, Seite 281) und LM (Load Memory, Seite 252).

Adressierungsmodus abfragen / ändern

AMODE Es wird der Adressierungsmodus ausgegeben, in dem CFS gegenwärtig läuft.

AMODE 31|24 Der Adressierungsmodus (Run-Mode) von CFS wird auf XS-31 oder NXS (24-bit Modus) gesetzt.

Hinweise:

Im AMODE31 wird bei der Ausführung verschiedener CFS-Kommandos bzw. Action-Codes (z.B. EDT, COMP) der benötigte virtuelle Speicher im oberen Adressraum angefordert. Dies bietet den Vorteil, daß die 16MB-Grenze nicht existiert und somit nahezu beliebig viel Speicher angefordert werden kann.

Beim Umschalten von AMODE31 auf AMODE24 is zu berücksichtigen, daß es im nachfolgenden Programmlauf zu Fehlersituationen kommen kann, die dadurch bedingt sind, daß im XS-Bereich angeforderter Speicher nicht mehr adressiert werden kann.

Drucker für Action-Code PDxx zuweisen

APD *device* Vordefinieren bzw. Ändern eines Druckerdevice-Namens, der in der PDFILE mit einem Mnemo-Code und dem virtuellen Device-Namen '????????' definiert ist. Falls einem virtuellen Device-Namen '????????' mit APD noch kein realer Drucker zugewiesen wurde, so wird der Druckername beim ersten Aufruf des entsprechenden Mnemo-Codes angefordert. Für weitere Informationen siehe CFS-Handbuch für Systemverwalter, Abschnitt "PDFILE".

Dateinamen in voller Länge anzeigen

CFN | NCFN Complete Filename. Die am Bildschirm angezeigte Dateienliste enthält die Datei- und Jobvariablennamen in der vollständigen (FSTAT-kompatiblen) Form, d.h. es werden stets die Cat-ID's und Benutzerkennungen der ausgewählten Dateien angezeigt. Der Systemverwalter kann den CFN-Modus zum Standard erklären (Änderung im Modul CFSMAIN).

NCFN No Complete Filename. Standardformat für die Ausgabe von Dateinamen in CFS. Es werden die Cat-ID's und Benutzerkennungen der selektierten Dateien nur dann in der Dateienliste ausgegeben, wenn dies zur Unterscheidung von anderen Dateien notwendig ist.

Connection-Hardcopy Taste

CH *t-bez* Connection Hardcopy. Mit der F3-Taste kann der Hardcopy-Modus für einzelne CFS-Masken angefordert werden. In Connections kann die F3-Taste durch eine Anwenderfunktion belegt sein. Aus diesem Grunde ist es möglich, mit dem Kommando CH für Connections eine beliebige F- bzw. K-Taste als Äquivalent zur F3-Taste festzulegen.

t-bez Tastenbezeichnung: K1, K3, ... ,K14 / F1, ... ,F20.
Standard: NONE (keine Taste zur Anforderung von Hardcopies für Einzelmasken in Connections).

Hinweise:

Beim Hardcopy einer Maske mit der durch das Kommando CH definierten Taste gehen die Eingaben verloren und müssen noch einmal eingegeben und mit ENTER gesendet werden.

Es ist zu beachten, daß bei Hardcopy mit der durch CH definierten Taste jede Maske in der Hardcopy-Datei auf einer neuen Seite erscheint, falls diese mit SPACE=E gedruckt wird. Im normalen Hardcopy-Modus (eingeschaltet durch das Kommando HC) werden nach Möglichkeit mehrere Masken auf einer Druckseite untergebracht.

BS2000-Prompt in Connections ändern

CP '*param*' In CFS-Connections wird als Eingabeaufforderung für BS2000-Kommandos eine frei definierbare Zeichenfolge festgelegt. Diese kann auch Variable, wie Datum, Uhrzeit, User-Id und Name des angeloggten Host enthalten. Standard: '/'

param Zeichenfolge für den BS2000-Prompt in Connections. *param* darf 60 Zeichen nicht überschreiten. Es findet keine Umsetzung von Kleinbuchstaben in die entsprechenden Großbuchstaben statt. Außer Konstanten können in *param* auch Variable verwendet werden. Diese werden zum Zeitpunkt der Ausgabe des Prompts durch die dann gültigen aktuellen Werte ersetzt. Der Name einer Variablen wird abgeschlossen durch ein Sonderzeichen, z.B. Blank oder Komma. Falls nach der Variablen ein Buchstabe oder

eine Ziffer als konstantes Zeichen ausgegeben werden soll, so ist der Name der Variablen durch einen Punkt abzuschließen. Der Punkt erscheint nicht im Text. Folgende Variablen sind möglich:

&CON	Nummer der Connection, in der sich der Benutzer gerade befindet.
&USER-ID	User-Id, unter der sich der Benutzer in der Connection angeloggt hat.
&LOGNAM	Jobname des LOGON-Kommandos für die Connection.
&HOST	Name des Hostrechners, unter dem der Dialogprozeß läuft.
&DATE	Tagesdatum in der Form tt.mm.yy
&TIME	aktuelle Uhrzeit in der Form hh:mm
&(*name*)	Inhalt der angegebenen Job- bzw. SDF-P Variablen.

Hinweise:

Der Connection-Prompt wirkt in allen Connections zu $DIALOG, sofern nicht mit der SDF-Option GUIDANCE= MIN | MED | MAX gearbeitet wird.

Falls das CP-Kommando in einer Startup-Datei enthalten ist und Variable verwendet werden, ist folgende Kommandofolge anzugeben, um Konflikte mit CFS-Prozedurvariablen zu vermeiden:

```
*SET PAR=#
CP ' ... '
*SET PAR=&
```

Falls das CP-Kommando in einer Startup-Datei enthalten ist, so können im Hexadezimalmodus des EDT auch Line-Mode Steuerzeichen zum Hervorheben von Textteilen in den String des CP-Kommandos eingesetzt werden. Siehe hierzu Manual "Makroaufrufe an den Ablaufteil", Makro VTCSET.

Beispiel: `CP 'x&CON:y &HOST $&USER-ID &TIME x->'`
x steht im obigen Beispiel für das Line-Mode Steuerzeichen X'14' (EM4) und y für das Zeichen X'1F' (EM2). Aufgrund des CP-Kommandos könnte in einer Connection 1 der folgende Prompt ausgegeben werden: `1: VAR2 $TSOS 10:22 ->`

Bei geschachtelten Connections (siehe Seite 352) wird stets der Connection-Prompt des zuletzt geladenen CFS wirksam.

Display-Datei nach jeder Maskeneingabe schließen

CA | NCA

Die Display-Datei wird nach jeder Maskeneingabe geschlossen. Während der Eingabepausen am Bildschirm ist die Datei geschlossen und kann von anderen Benutzern lesend oder schreibend eröffnet werden.
Standard: NCA

Hinweise:

Beim Modifizieren von Dateien (Modify) bleibt der CA-Modus ohne Wirkung, d.h. die zur Änderung freigegebene Datei bleibt bis zum Verlassen des Modify-Modus gegen Schreibzugriff durch andere Benutzer geschützt.

Der CA-Modus ist ebenfalls ohne Wirkung bei der Einzelsatzdarstellung (Single Record-Modus, SR), sowie beim Display von Banddateien.

Der CA-Modus wird mit dem Verlassen des Display-Modus nicht ausgeschaltet, sondern wirkt auch für alle folgenden Display-Aktionen solange bis er durch das Kommando NCA wieder zurückgenommen wird.

Eingaben in Großbuchstaben umwandeln

CAP CFS-Editor: Beim Modifizieren werden Kleinbuchstaben vor dem Zurückschreiben in die Datei in die entsprechenden Großbuchstaben umgewandelt.

Eingabe von Dateidokumentationen (Action-Code IM): Der eingegebene Text wird in Großbuchstaben umgesetzt.

Suche-Kommando, User Option FIND, Variable Action ONXFIND: Suchargumente ([C]'*item*') werden in Großbuchstaben umgesetzt.

Anstelle von CAP kann auch das Kommando LOWO (LOW Off) angegeben werden. Die Umwandlung eingegebener Kleinbuchstaben in Großbuchstaben wird unterbunden durch das Kommando LOW.

Action-Code EDT: Beim Einlesen einer neuen Datei/eines neuen Bibliothekselements wird der zuvor evtl eingeschaltete LOWER ON-Modus des EDT zurückgesetzt (LOWER OFF).

Bildschirmmasken vor der Ausgabe komprimieren

CSO [STAT] | NCSO Compress Screen Output on/off. Von CFS ausgegebene Bildschirmmasken und Formatausgaben in Connections werden vor der Darstellung am Bildschirm, d.h. bevor sie über die Leitung gesendet werden komprimiert. Die Komprimierung besteht darin, daß aufeinanderfolgende gleiche Zeichen durch spezielle Steuerzeichen (WDH-Folgen) ersetzt werden. Um optimale Komprimierungsraten zu erzielen, wird die Dateienliste von CFS in einem leicht modifizieren Format dargestellt. Vor der Beendigung von CFS wird eine Meldung mit der Kompressionsstatistik ausgegeben.

STAT Es wird der momentane Stand der Kompressionsstatistik in des Systemzeile des Bildschirms ausgegeben.

NCSO No Compress of Screen Output. Die Bildschirmkomprimierung wird ausgeschaltet.
Standard: NCSO.

Überprüfung des Elementtyps bei Aufnahme von Elementen in PLAM-Bibliotheken

CT | CTQ | NCT Check Types/Check Types and Query/No Check of Types. Es erfolgt eine Überprüfung des Elementtyps bei Aufnahme von Dateien in PLAM-Bibliotheken.

CT Es sind nur die LMS-kompatiblen Elementtypen C/D/H/J/L/M/P/R/S/X erlaubt.
CTQ Bei Angabe nicht LMS-kompatibler Bibliothekstypen wird der Benutzer gefragt, ob er eine Aufnahme unter dem angegebenen Typ wirklich wünscht.
NCT Keine Typ-Überprüfung bei Aufnahme von Dateien in PLAM-Bibliotheken.
Standard: CTQ.

Age-Angabe in Datumsform

DATE Altersangaben, insbesondere die Angaben in der AGE-Spalte der Dateienliste, werden nicht als Anzahl von Tagen, sondern in Form eines Datums (dd.mm.yy) angezeigt. In der User Option-Spalte kann das Datum aus Platzgründen nur in der Form YMMDD angezeigt werden. Y ist hierbei die letzte Stelle des Jahres (z.B. 6 für 1996). Durch das Kommando NDATE [oder AGE] kann dieser Modus wieder auf die Standardeinstellung zurückgesetzt werden. Standard: AGE.

Dokumentationsdatei definieren

DOC Feld DOCUMENTATION der Selektionsmaske anzeigen.

DOC *datei* Feld DOCUMENTATION direkt mit dem angegebenen Inhalt belegen.

Für weitere Informationen wird auf die Beschreibung des Feldes DOCUMENTATION Seite 74 verwiesen.

DO-Parameter Maske bei Prozeduren ausgeben

DP | NDP Do-Parameter mask/No Do-Parameter mask.

DP Bei einem BS2000 DO-Kommando, das im Kommandofeld der Dateienliste eingegeben wurde bzw. bei DO-Kommandos in Connections, wird die PROC-Anweisung der DO-Prozedur analysiert. Wird dabei festgestellt, daß DO-Parameter existieren, die noch nicht mit einem Wert versorgt sind, so wird dem Benutzer eine Parameter-Maske angeboten. In ihr können die noch fehlenden DO-Parameter nachträglich angegeben werden.
Standard: DP.

Hinweis:

Bei DO-Kommandos in Connections wird die Parameter-Maske auch im DP-Modus nicht ausgegeben, falls der Nebenprozeß unter einer anderen Benutzerkennung als der Grundprozeß von CFS läuft und die angesprochene DO-Prozedur nicht shareable ist bzw. falls die Connection auf einem anderen als dem eigenen Host-Rechner eröffnet wurde.

Display used resources

DU [M | CPU | WSI | IO]

In der Maske der Dateienliste wird anstelle der Uhrzeit (*hh*:*mm*:*ss*), eine Angabe zu den im letzten Transaktionsschritt verbrauchten Systemresourcen gemacht.

M Display Used Memory. Es wird die Größe des belegten Klasse-6 Speichers im Adressbereich unterhalb 16MB in K (Kilobyte) angezeigt.

CPU Display Used CPU-Time. Es wird die seit der letzten Bildschirmausgabe verbrauchte CPU-Zeit angezeigt.

WSI Display Used Working Set Integral. Es wird der Wert des Working Set Integrals seit der letzten Bildschirmausgabe angezeigt. Das Working Set Integral ist definiert als das Produkt aus Hauptspeicherseiten * verbrauchte Zeit.

IO — Display Used IO's. Es wird die Anzahl der seit der letzten Bildschirmausgabe angefallenen Platten-IO's angezeigt.

NDU — Anstelle der verbrauchten Systemresourcen wird die Uhrzeit (*hh*:*mm*:*ss*) angezeigt. Standard: NDU.

Duplicate Key-Modus im CFS-Editor

DUPK | NDUPK — DUPKEY-Modus Ein/Aus. Bei der Bearbeitung von ISAM-Dateien sind gleichlautende Schlüssel möglich/nicht möglich. Bei Move und Copy-Operationen können im DUPK-Modus ISAM-Sätze mit gleichen Schlüsseln erzeugt werden (siehe Seite 328). Im DUPK-Modus können auch mit dem IR-Kommando (Insert Records) Sätze mit gleichlautenden Schlüsseln erzeugt werden (insbesondere mit der K-Option: schlüsselgerechte Einfügung). Der DUPK-Modus wirkt schließlich auch beim W-Kommando aus ISAM-Dateien: Es können die gleichen Datensätze mehrmals in die Write-Datei geschrieben werden.
Standard: NDUPK.

Automatische Kommandos nach Aufruf des EDT

EDTINIT [*cmd;cmd; ...*]

Nach dem Einlesen eines Datenobjekts mit dem Action-Code EDT[*n*] oder dem Kommando EDT, werden die angegebenen EDT-Kommandos ausgeführt.

cmd — Ein oder mehrere EDT-Kommandos. Mehrere Kommandos sind mit dem Zeichen ';' zu trennen. Beispiel: `EDTINIT INF=ON,LOWER ON;PAR EDIT-FULL=ON`

Standard: Es werden keine benutzerdefinierten EDT-Kommandos ausgeführt.

Hinweise:

Beim gleichzeitigen Einlesen mehrerer Datenobjekte in verschiedene Ebenen des EDT (Action-Code EDT[*n*]) werden die in EDTINIT angegebenen Kommandos in jeder EDT-Ebene einzeln ausgeführt.

Das Kommando EDTINIT ohne Parameter setzt die Initialisierungskommandos für den EDT zurück.

Erase with Retain of Tempfiles

ERT | NERT — Erase Retain Tempfiles/Erase with No Retain of Tempfiles.

ERT — Der ERT-Modus hat zur Folge, daß die mit dem Action-Code E gelöschten Datenobjekte zunächst in temporäre Dateien umkatalogisiert bzw. in temporäre Bibliotheken übertragen werden. Diese temporären Dateien werden vom BS2000 bei Prozeßende gelöscht. Versehentlich vom Benutzer mit dem Action-Code E markierte und gelöschte Daten sind damit noch bis Prozeßende physisch als Dateien vorhanden (zugreifbar durch Kommando NP# bzw. NP STAJV #).

Die Namen der erzeugten Temporärdateien werden gebildet, indem beim CAT-Kommando den Namen der zu löschenden Dateien das Tempfile-Zeichen (normalerweise #) vorangestellt wird. Daraus ergibt sich, daß der ERT-Modus beim expliziten Löschen von temporären Dateien unwirksam ist.

Bibliothekselemente werden vor dem Löschen unter dem gleichen Namen in eine temporäre Bibliothek kopiert und sind dort bis Prozeßende noch zugreifbar. Der Name der temporären Bibliothek wird gebildet, indem dem Namen der Bibliothek aus der Elemente gelöscht werden, das Tempfile-Zeichen # vorangestellt wird.

Die ERT-Option **wirkt nicht**, falls

- Dateien mit den Action-Codes ED (Erase...,Data), EC (Erase..., Catalog) oder EN (Erase No Retain Tempfiles) gelöscht werden,
- Dateien auf Privatplatten mit mehr als 5000 beschriebenen Seiten mit Action-Code E gelöscht werden,
- Dateien mit einer PVS-Bezeichnung im Feld USER-ID selektiert wurden und diese mit Action-Code E gelöscht werden,
- unter TSOS Dateien einer fremden Kennung selektiert wurden und diese mit Action-Code E gelöscht werden,
- PLAM-Bibliothekselemente mit dem Action-Code EA (Erase All delta versions) gelöscht werden.

Gelöschte Datenobjekte werden automatisch aus der aktuellen Dateienliste entfernt (impliziter Action-Code '-'). Mit dem Kommando YANK können diese unsichtbaren Zeilen der Dateienliste wieder sichtbar gemacht werden.

Die ERT-Option ist standardmäßig eingeschaltet.

NERT — Die ERT-Option läßt sich ausschalten mit dem Kommando NERT (No Erase Retain Tempfiles) bzw. über die SET-Maske.

Hardcopy-Modus einschalten

HC [*n*] [*datei*] [, L] [, T] [, B] [, E] [, O] [, A] [, F] [, G] [, I] [, S] [, R]

Hardcopy-Modus für die CFS-Grundebene/für Connection *n* einschalten.
Bei eingeschaltetem Hardcopy-Modus werden die anfallenden Bildschirm Ein-/ Ausgaben (insbesondere Masken) bildschirmgerecht in einer Datei protokolliert.

Alle Operanden des HC-Kommandos sind wahlweise und können weggelassen werden. Es treten dann die entsprechenden Standardannahmen in Kraft.

Der Hardcopy-Modus kann beliebig oft ein- und wieder ausgeschaltet werden (Kommando NHC siehe unten).

Ohne den Hardcopy-Modus über das HC-Kommando einzuschalten, können einzelne CFS-Masken auch protokolliert werden, indem sie in der CFS-Grundebene mit der F3-Taste und in einer Connection mit einer vorher definierten Hardcopy-Taste (siehe Kommando CH, Seite 470) abgesendet werden. Es ist jedoch zu beachten, daß jede mit **F3** protokollierte Maske im ausgedruckten Listing auf einer neuen Seite erscheint. Im normalen Hardcopy-Modus (HC-Kommando) werden nach Möglichkeit mehrere Masken auf einer Druckseite untergebracht.

Das Absenden einer CFS-Maske mit **F3** wird, abgesehen von der Protokollierung wie ENTER behandelt. Beim Hardcopy einer Maske in einer Connection mit der durch das Kommando CH definierten Datenübertragungstaste gehen die Eingabedaten verloren und müssen noch einmal eingegeben und mit der ENTER-Taste abgesendet werden.

Das Kommando HC wirkt nur für den Basisprozeß von CFS und nicht für evtl. durch OC eröffnete Connections. Für diese muß ein HC*n*-Kommando angegeben werden. Der Hardcopy-Modus für eine Connection kann bereits vor dem Eröffnen einer Connection durch das OC-Kommando eingeschaltet werden. Innerhalb der Connection kann der Hardcopy-Modus durch das HC-Kommando angefordert werden.

Für eine ausführliche Beschreibung aller Operanden des HC-Kommandos wird auf Seite **439** und folgende (Hardcopy) verwiesen.

NHC | NHC *n* — Hardcopy-Modus für CFS-Grundebene / für Connection *n* ausschalten.

Dateiinformationssystem: Standard-Format für die Erfassung von INF-Texten

I FORM=*f* — Mit diesem Kommando kann ein benutzerindividuelles Format *f* im zentralen Formatkatalog $TSOS.CFS.INF.FORMAT definiert werden, das standardmäßig bei der Neuerfassung von Dokumentationen verwendet wird (Action-Codes IM/IA). Für weitere Informationen siehe Seite **421** und folgende (Datei-Informationssystem).

Add in Bibliotheken: Versionsgeführte Elemente anlegen

IV | NIV — Install Version/No Install Version.

IV — Bei Add in FMS-Bibliotheken wird ein Element bei erstmaliger Aufnahme in die Bibliothek als versionsgeführtes Element eingerichtet (Delta-Speicherung).

NIV — Falls das neu aufzunehmende Element in der FMS-Bibliothek noch nicht existiert, so wird dieses als nicht versionsgeführtes Element eingerichtet.
Standard: NIV

Job-Report aktivieren / deaktivieren

JR [*jr-id*] — Job-Report System aktivieren.

NJR — Job-Report Einrichtung deaktivieren
Standard: JR.

Für weitere Informationen zu JR und allen mit JR beginnenden Kommandos wird auf Seite **410** und folgende verwiesen.

Job-Report Nachricht bei Beendigung von File-Transfer Aufträgen

JRFT — Das JRFT-Kommando bewirkt, daß alle nachfolgend über CFS gestarteten File-Transfer Aufträge, bei denen in der Maske der Parameter PROTOCOL-LISTING : YES gesetzt wurde, mit einer Monitor-Jobvariablen verknüpft werden. Dies hat zur Folge, daß bei Beendigung der FT-Aufträge das Protokoll über den Job-Report von CFS angezeigt wird.

NJRFT File-Transfer Aufträge werden nicht mit einer Monitor-Jobvariablen und somit auch nicht mit dem Job-Report von CFS verknüpft. Das gleiche gilt auch, falls in der File-Transfer Maske von CFS der Parameter PROTOCOL-LISTING auf dem Standardwert NO belassen wurde.
Standard: JRFT bei PROTOCOL-LISTING : YES

Job-Report Parameter setzen

JREP *enter-params* Job-Report Enter-Parameter. Für alle mit dem Job-Report System gestarteten Aufträge werden im Enter-Kommando die angegebenen Parameter verwendet.

enter-params Es können alle im BS2000 erlaubten Enter-Parameter angegeben werden. Diese Parameter gelten dann für alle nachfolgenden E-Kommandos. Die durch das JREP-Kommando vorbelegten Enter-Parameter können im E-Kommando und/oder in der Parameter-Maske für DO-Prozeduren noch durch weitere Enter-Parameter ergänzt werden.

Die Enter-Parameter können auch in der JRP-Maske (Kommando JRP, siehe unten) eingegeben werden.

JRF <u>SYSOUT</u> | SYSLST | *jr-file*

Job-Report-File definieren. Hiermit wird festgelegt, welche Datei bei Jobende angezeigt werden soll. Die JRF-Angabe gilt für alle nachfolgend gestarteten Jobs. Die Job-Report-File kann auch als Parameter beim E-Kommando oder im Kommando JRP angegeben werden.
Standard: SYSOUT.

JRIC *cmd* Job-Report Initial Command. Für *cmd* ist jedes in der Job-Report Maske erlaubte Kommando möglich (+/-/++/S,...). Das angegebene Kommando wird im Job-Report bei Beendigung eines Enter-Prozeßes automatisch ausgeführt (Default-Kommando). Das JRIC kann auch im Kommando JRP angegeben werden.
Standard: kein Initial Command

JRID *jr-id* Benutzerindividuelle Job-Report-Identifikation definieren. Die JRID kann auch im Kommando JRP angegeben werden.
Standard: Abrechnungs-Nr.

JRSH | <u>NJRSH</u> Option SHARE=YES oder NO für die vom JR-System erstellten Jobvariablen und Dateien. Standard: NJRSH (SHARE=NO)

JRP Job-Report Parameters. Es wird eine Maske ausgegeben, in der die aktuellen Werte für JRID, JRF, JRIC, JREP und JRSH angezeigt werden. Die Werte können verändert werden und gelten für alle nachfolgenden Job-Report-Aktionen.

Letztes Kommando nicht löschen

KC | <u>NKC</u> Keep Command /do Not Keep Command.

KC Das zuletzt eingegebene Kommando wird im Kommandofeld nicht gelöscht.
NKC Das zuletzt eingegebene Kommando wird bei korrekter Ausführung gelöscht.
Standard: NKC

Zum Thema "Letztes Kommando wiederholen", siehe auch Seite 211 (Gedächtnis der eingegebenen Kommandos).

Last Page als Merkmal der Dateigröße

LASTP

In der ersten Spalte der Dateienliste wird als Merkmal für die Größe der Datei der Wert Last Page angezeigt, d.h. die Anzahl der tatsächlich beschriebenen PAM-Blöcke.
Standard: LASTP.

Eingaben [nicht] im Kommandogedächtnis ablegen

[**N**]**LIM** [SEL|CMD|CON]

Log Input in Memory (LIM) / do Not Log Input in Memory (NLIM).

LIM

Alle Benutzereingaben (Eingaben in der Selektionsmaske, Kommandos, Line-Mode Eingaben in Connections) werden im internen Kommandogedächtnis protokolliert (Standard).

SEL

Das Gedächtnis für Eingaben in der Selektionsmaske wird aktiviert.

CMD

Das Gedächtnis für Eingaben in der Dateienliste (Feld "COMMAND :") wird aktiviert.

CON

Das Gedächtnis für Eingaben in Connections wird aktiviert.

NLIM

Alle nach diesem Kommando getätigten Eingaben werden nicht im internen Kommandogedächtnis protokolliert und stehen somit bei einer späteren DUE2-Anforderung nicht zur Verfügung. Das Kommando NLIM ist besonders in CFS-Prozeduren sinnvoll, da hiermit die von der Prozedur erzeugten Eingaben nicht in das Kommandogedächtnis aufgenommen werden.

SEL

Kein Gedächtnis für Eingaben in der Selektionsmaske.

CMD

Kein Gedächtnis für Eingaben in der Dateienliste (Feld "COMMAND :").

CON

Kein Gedächtnis für Eingaben in Connections.

Hinweise:

Im Initialisierungsmodul CFSMAIN kann vorgegeben werden, welche Eingaben in das Kommandogedächtnis aufgenommen werden sollen.

Für weitere Informationen zum Thema Kommandogedächtnis siehe Seite **211**.

Kleinbuchstaben nicht in Großbuchstaben umwandeln

LOW

CFS-Editor: Beim Modifizieren wird keine Umwandlung der eingegebenen Kleinbuchstaben in Großbuchstaben vorgenommen.

Eingabe von Dateidokumentationen (Action-Code IM): Der Text wird in der eingegebenen Groß-/Kleinschreibung übernommen.

Suche-Kommando, User Option FIND, Variable Action ONXFIND: Es erfolgt keine Umwandlung von Kleinbuchstaben im Suchargument in Großbuchstaben. Diese Funktion kann jedoch auch durch Eingabe des Suchargumentes in der Form L'....' erreicht werden.

Action-Code EDT: Beim Einlesen einer neuen Datei/eines neuen Bibliothekselements wird der LOWER-Modus des EDT unverändert gelassen. Ein bei der Bearbeitung der letzten Datei eingegebenes LOWER ON-Kommando bleibt damit wirksam.
Standard: CAP

Speicherbereich in CFS modifizieren

M L'*addr*', C'*string*' | X'*string*'

Mit diesem Kommando können im geladenen Programm CFS bestimmte Speicherbereiche mit einem vom Standard abweichenden Wert überschrieben werden. Das Modify-Kommando ist ein Ersatz für das AID-Kommando %M C'...' INTO V'...'. Das Modify-Kommando sollte nur zum Ändern von CFS-internen Parametern verwendet werden, für die kein eigenes Kommando existiert.

L'*addr*' Adresse der zu modifizierenden Stelle im geladenen Programm CFS. Die Adressen bestimmter Parameter in CFS können aus dem Quellprogramm CFSMAIN ermittelt werden. Dem Assembler-Befehl MVI 33(R1),C'Y' entspricht in der CFS-Syntax z.B. das Kommando M L'2021',C'Y' (L'2021' = L'2000' + 33). Die in CFSMAIN angesprochenen Adressen CFS-interner Parameter werden sich auch in zukünftigen CFS-Versionen nicht ändern.

C'*string*' | X'*string*' An die angegebene Adresse zu übertragender Inhalt.

Beispiele:

```
M L'2031',X'00'
```
Mit dieser Anweisung wird der Inhalt des Bytes an der Adresse L'2031' auf den Wert X'00' gesetzt. Damit wird erreicht, daß beim Hardcopy von Masken in eine druckaufbereitete Datei Kleinbuchstaben von CFS nicht in Großbuchstaben umgesetzt werden.

```
M L'22E5',C'CFS.JRSAVE.name     '
```
Mit dieser Anweisung wird der Inhalt des Speicherbereichs ab Adresse L'22E5' mit der Character-Zeichenfolge 'CFS.JRSAVE.name ⊔ ⊔ ⊔ ' gefüllt. Damit wird ein von der Standard JRSAVE-Datei abweichender Name vereinbart, z.B. für zwei Benutzer unter der gleichen Kennung.

Benutzeranfragen wegen Überschreiben u.ä.

QE | NQE Query/No Query on Erase.
Um ein unbeabsichtigtes Löschen zu vermeiden, kann der Benutzer mit QE veranlassen, daß jeder Löschvorgang mit dem Action-Code E (Erase) vor der Ausführung eigens bestätigt werden muß. Standard: NQE (No Query on Erase).

Im Zusammenhang mit dem Löschen von Datenobjekten wird auch auf den ERT-Parameter von CFS hingewiesen (ERT = Erase with Retain of Tempfiles).

QA | NQA Query/No Query on Add.
Vor der Ausführung eines ADD in ein bereits existierendes Bibliothekselement wird der Benutzer gefragt, ob er das bereits bestehende Element auch tatsächlich überschreiben möchte: ".... IS ALREADY IN LIBRARY - OVERWRITE ? (Y/N/YA/NA)". Während der Ausführung einer Variablen Action bewirkt die Antwort YA (Yes,All) das Überschreiben des aktuellen Bibliothekselements. Für weitere Elemente werden keine Rückfragen mehr gestellt. Es wird automatisch die Antwort Y angenommen. Analoges gilt für die Antwort NA (No,All). Der YA/NA-Modus wird mit Beendigung der Variablen Action wieder außer Kraft gesetzt.
Standard: NQA (No Query on Add).

QC | NQC — Query/No Query on Copy.
Vor der Ausführung eines Copy in eine bereits existierende Datei/JV/Bibl.-Element wird der Benutzer gefragt, ob er das bestehende Datenobjekt tatsächlich überschrieben werden soll: "... IS ALREADY IN CATALOG - OVERWRITE ? (Y/N/YA/NA)". Während der Ausführung einer Variablen Action bewirkt die Antwort YA (Yes, All) das Überschreiben des Datenobjekts. In weiteren Verlauf werden keine Rückfragen mehr gestellt. Es wird automatisch die Antwort Y angenommen. Analoges gilt für die Antwort NA (No,All). Der YA/NA-Modus wird mit Beendigung der Variablen Action wieder außer Kraft gesetzt.
Standard: NQC (No Query on Copy).

QR | NQR — Query/No Query on EDT-Return.

QR — Kommando EDT/EDL: Nach der Rückkehr aus dem EDT wird der Benutzer gefragt, ob der virtuelle EDT-Speicher gelöscht werden soll (@D).
Action-Codes EDT/EDL: Nach erfolgreicher Ausführung der Action UPD wird der virtuelle EDT-Speicher automatisch gelöscht.

NQR — Virtueller EDT-Speicher wird nach Action-Code UPD nicht gelöscht; keine Frage an Benutzer wegen @D nach Rückkehr aus dem Kommando-EDT.
Standard: QR.

QS | NQS — Query/No Query on Select.
Vor der Ausführung eines Select auf eine bereits bestehende Datei wird der Benutzer gefragt, ob die Datei tatsächlich überschrieben werden soll: ".... IS ALREADY IN CATALOG - OVERWRITE ? (Y/N/YA/NA)". Während der Ausführung einer Variablen Action bewirkt die Antwort YA (Yes,All) das Überschreiben der Datei. Bei weiteren Dateien werden keine Rückfragen mehr gestellt. Es wird automatisch die Antwort Y angenommen. Analoges gilt für die Antwort NA (No,All). Der YA/NA-Modus wird mit Beendigung der Variablen Action wieder außer Kraft gesetzt.
Standard: QS (Query on Select).

QT | NQT — Query/No Query on Term.
Vor Beendigung des CFS wird der Benutzer gefragt: "DO YOU REALLY WANT TO TERMINATE CFS ? (Y/N)". Standard: NQT.

QL | NQL — Query/No Query on Lock.
Vor dem Sperren des Bildschirms durch das Kommando LOCK wird der Benutzer gefragt: "Do you really want to lock your screen ? (Y/N)". Standard: NQL.

Q | NQ — Query/Non Query.

Q — gleichbedeutend mit QE;QA;QC;QT;QR;QL.

NQ — gleichbedeutend mit NQE;NQA;NQC;NQT;NQR;NQL.

Spaltenzähler einblenden

SC | SCO | NSC — SCale/SCale Off. In der oberen Bildschirmhälfte wird ein "Lineal" mit einem Spaltenzähler angezeigt. Das Lineal ist im Display-Modus und bei der Anzeige der Dateienliste aktiv. Alternativ zu SC/SCO können auch die Kommandos OL/NOL (Orientation Line/No Orientation Line) angegeben werden.
Standard: SCO (kein Lineal anzeigen).

Connection-Status in Zeile 25 einblenden

SCL | NSCL

Show Connection Line.
Befindet sich der Benutzer in einer Connection, so wird am Bildschirm in Zeile 25 eine Übersicht der wichtigsten Connection-Parameter angezeigt, wie z.B. die Nummer der aktuellen Connection, der Anwendungsname, der virtuelle Terminalname, der Hostname und die Benutzerkennung (nur bei $DIALOG).
Standard: NSCL

Hinweis:

Durch die Connection-Zeile erhält der Benutzer jederzeit den Überblick über die gerade aktive Connection. Dies ist besonders bei UTM- und DCAM-Anwendungen von Vorteil, da hier nicht wie bei Verbindungen zu $DIALOG ein Connection-Prompt (siehe Kommando CP) ausgegeben werden kann.

PLAM-Bibliotheken: Age aus "System-Date" bestimmen

SDAT

PLAM-Bibliothekselemente: Die Age-Angaben in der Elementliste werden aufgrund des vom Benutzer nicht beeinflußbaren System-Datums (=Datum des letzten Updates) errechnet. Standard: UDAT

Hinweis:

Beim Reorganisieren von PLAM-Bibliotheken wird das System-Datum aller Elemente auf den aktuellen Tag gesetzt. Das User-Datum bleibt unverändert. Gleiches gilt auch, falls mit LMS REP-Anweisungen in ein Modul vom Typ R eingebracht werden.

Trennungszeichen für CFS-Kommandos ändern

Im Kommandofeld von CFS können mehrere Kommandos nacheinander ausgeführt werden. Die einzelnen Kommandos werden getrennt durch das Separatorzeichen ";".

Das Separatorzeichen wird außerdem verwendet zur

- Trennung von Schlüsselwortparametern in *001/*002-Sätzen (siehe Seite **485** und folgende (CFS-Prozedursprache),
- Trennung der Schlüsselwortparameter in den Kommandos NP (New Parameters) und AL (Append List),
- Aneinanderreihung mehrerer User Options im entspr. Feld der Selektionsmaske bzw. im Kommando OU (Standard User Options).

Das Trennungszeichen kann umdefiniert werden durch das Kommando SEP.

SEP *x*

x ist das neue Separatorzeichen. Das Kommando SEP darf nicht mit anderen Kommandos verkettet eingegeben werden.

SEP ?

zeigt das aktuelle Separatorzeichen an. Siehe auch das entsprechende Feld in der SET-Maske von CFS.

Standard Sort Option für Dateienliste festlegen

SO { (*c*,*l*) | AGE | NAME | SIZE | USER | NONE } [, A | D]

Die Dateienliste wird standardmäßig nach Namen aufsteigend sortiert angezeigt (Standard Sortierkriterium). Mit dem Kommando SO hat der Benutzer die Möglichkeit, ein anderes Feld als den Dateinamen als primäres Sortierkriterium, sowie die Reihenfolge der Sortierung (auf-/absteigend) festzulegen. Felder mit dem gleichen primären Sortierkriterium werden in zweiter Instanz automatisch nach Namen aufsteigend sortiert. Mit SO wird das im Feld SORT OPTION der Selektionsmaske angezeigte Standard Sortierkriterium vorbelegt. Eine Eingabe im Feld SORT OPTION (siehe Seite 72) wirkt nur temporär und überschreibt nicht das durch SO festgelegte Standard Sortierkriterium. Bezüglich der nachträglichen Sortierung der Dateienliste siehe auch Kommando SORT auf Seite 282.

AGE — Die Dateienliste wird gemäß dem Inhalt in der AGE-, bzw. DATE-Spalte sortiert. Bei PLAM-Bibliotheken wird die Uhrzeit der letzen Änderung (TIME) im Rahmen der Sortierung mit berücksichtigt.

NAME [(*c*,*l*)] — Die Dateienliste wird bezüglich der Namen der Datenobjekte sortiert (Standard). Über den Zusatz (*c*,*l*) kann ein Teil des Namens als Sortierfeld festgelegt werden. *c* gibt dabei die Spalte und *l* die Länge des Namensteils an.
Standard: NAME(1,44).

Hinweis:
Bei PLAM-Bibliotheken kann durch die Sort Option NAME(1,44) eine Sortierung erreicht werden, bei der Elemente mit gleichen Namen und verschiedenen Typ-Bezeichnungen aufeinander folgen.
Durch die Sort Option NAME ohne Zusatz wird bei PLAM-Elementen der Elementtyp bei der Sortierung der Elementnamen mit berücksichtigt. Es ergibt sich in diesem Fall ein anderes Sortierergebnis als bei der ansonsten gleichwertigen Angabe NAME(1,44).

SIZE — Die Dateienliste wird gemäß dem Inhalt der ersten Spalte (LASTP, bzw. ALLOC bei Dateien und SPACE bei Bibliotheken) sortiert.

USER — Falls bei der Selektion der Dateienliste eine USER OPTION angegeben wurde, so wird durch die Sortieroption USER eine Sortierung gemäß dem in der User Option-Spalte enthaltenen Wert vorgenommen. Falls keine USER OPTION angegeben wurde, so bleibt diese Sortieroption ohne Wirkung.

NONE — Die Dateien-/Elementliste wird unsortiert ausgegeben.

A — Aufsteigende Sortierreihenfolge (Standard).

D — Absteigende Sortierreihenfolge.

SO? — Es wird die aktuell gültige Sortieroption angezeigt.

Passwort für CFS-Benutzungssperre vorgeben

SPL [*passw*] — Set Password for Lock. Es wird ein Passwort für ein später folgendes LOCK-Kommando definiert. Wird *passw* im SPL-Kommando nicht angegeben, so kann der Benutzer das Passwort in einem dunkel gesteuerten Feld eintragen.

PLAM-Bibliotheken: Age aus "User-Date" bestimmen

UDAT

PLAM-Bibliothekselemente: Die Age-Angaben in der Elementliste werden aufgrund des User-Datums errechnet.
Standard: UDAT.

Hinweise:

Beim Reorganisieren von PLAM-Bibliotheken wird das System-Datum aller Elemente auf den aktuellen Tag gesetzt. Das User-Datum wird beim Reorganisieren nicht verändert.

Bei Aufnahme einer Datei als Element in eine PLAM-Bibliothek wird von CFS das User-Datum des Bibliothekselements auf das Change Date der Datei gesetzt. Dies kann zur Folge haben, daß im UDAT-Modus neu aufgenommene Dateien mit einem Age-Wert > 0 in der Bibliotheksliste von CFS erscheinen.

Userlib auf andere Bibliothek zuweisen

UL *bibliothek*

Die angegebene Bibliothek wird fortan als CFS-Userlib verwendet. Eine Information über die aktuell zugewiesene Userlib und das Zuweisen einer anderen Bibliothek kann auch in der Maske des SET-Kommandos erfolgen.
Standard: CFS.USERLIB (eigene Benutzerkennung)

Für weitere Informationen zum Thema CFS-Userlib wird auf Seite **541** und folgende (Userlib) verwiesen.

Standard User Option für die Selektionsmaske definieren

UO [*u-opt-1*] [; [*u-opt-2*]] [; [*u-opt-3*]] [; [*u-opt-4*]]

Die angegebene User Option bzw. die angegebenen User Options werden fortan als Standardwert in der Selektionsmaske von CFS eingetragen. Eine Information über die aktuelle Standard User Option und das Zuweisen einer anderen Standard User Option kann auch in der Maske des SET-Kommandos erfolgen.
Standard: keine standardmäßige User Option.

Hinweise:

User Options bewirken, daß zusätzlich zu den von CFS standardmäßig angezeigten Dateimerkmalen noch ein oder mehrere weitere, vom Benutzer frei wählbare Merkmale in der Dateienliste ausgegeben werden. Durch Angabe einer Auswahlbedingung können Datenobjekte bezüglich dieser Merkmale auch selektiert werden. Mehrere User Options sind durch Semikolons zu trennen. Es können maximal vier User Options angegeben werden.

Das Kommando UO ohne Operanden macht die zuletzt definierten Standard User Options ungültig.

Das Kommando UO darf nicht mit weiteren Kommandos durch Semikolon ";" verkettet werden.

Der Wert für die als erstes angegebene User Option wird in einer eigenen User Option Spalte in der Dateienliste eingetragen. Die Werte für eine evtl. angegebene zweite, dritte und vierte User Option überschreiben in der Dateienliste die Spalten PASS, AGE und LASTP (SPACE).

Zwei aufeinanderfolgende Semikolons ";;" bewirken, daß die entsprechende Spalte der Dateienliste nicht mit einer User Option überschrieben wird. Siehe hierzu das zuletzt angegebene Beispiel.

Eine mit UO festgelegte User Option wirkt erst bei der nächsten Selektion. Um die neue User Option für die aktuelle Liste wirksam werden zu lassen, kann das Kommando NP* angewendet werden. NP* zeigt die neue User Option jedoch dann nicht an, falls in der aktuellen Selektion bereits eine andere User Option definiert war.

Beispiele:

`ALLO2;ALCNT;EXTNT;ALLOC`
Hiermit wird die User Option Spalte mit der Größe der Sekundärzuweisung (ALLO2) gefüllt. Die Spalte PASS zeigt die Anzahl der Sekundärzuweisungen (ALCNT), die Spalte AGE die Anzahl der Extents (EXTNT) und die Spalte LASTP die Anzahl Seiten, die der Datei zugewiesen wurden (ALLOC).

`ALCNT;EXTNT;;ALLOC`
Hiermit wird die User Option Spalte mit der Anzahl der erfolgten Sekundärzuweisungen (ALCNT) und die Spalte PASS mit der Anzahl der Extents (EXTNT) gefüllt. Die Spalte AGE wird nicht überschrieben (;;). Die Spalte LASTP wird mit dem der Datei zugewiesenen Platz (ALLOC) gefüllt.

Standard User-ID für die Selektion definieren

UID [*:x:* | *:*:* | *user-id* | *:x: user-id* | *:*: user-id*]

Die angegebene User-Id wird fortan als Standardwert in der Selektionsmaske von CFS eingetragen. Eine Information über die aktuelle Standard User-Id und das Zuweisen einer anderen Standard User-Id kann auch in der Maske des SET-Kommandos erfolgen.

user-id Selektion von Dateien im Default-Pubset der angegebenen Benutzerkennung. Die Benutzerkennung ist ohne $ und nachgestellten Punkt anzugeben.

Standard: keine standardmäßige User-Id.

Hinweise:

Das Kommando UID ohne Operanden macht die zuletzt definierte Standard User-Id ungültig.

Eine mit UID festgelegte User-Id wirkt erst bei der nächsten Selektion.

Feld VARIABLE ACTION in der Selektionsmaske anzeigen/besetzen

VAR Feld VARIABLE ACTION der Selektionsmaske anzeigen.
Der Benutzer hat hier zwei Zeilen am Bildschirm zur Verfügung, in die er die gewünschte Variable Action eintragen kann.

VAR ONX/& *var-act*
Gleiche Wirkung wie ONX/& *var-act* : Das Feld VARIABLE ACTION wird mit dem angegebenen Inhalt gefüllt.

19. Prozedursprache

Übersicht

CFS bietet dem Benutzer die Möglichkeit, eine Folge von Kommandos/Selektionseingaben auch im prozedurgesteuerten Modus auszuführen. Die auszuführenden Kommandos können enthalten sein in einer

- BS2000 DO-/Enter-Prozedur
- CFS-Kommandoprozedur
- Startup-Datei von CFS

Die im folgenden Kapitel beschriebene Prozedursprache von CFS gilt in gleicher Weise für den Aufruf von CFS in BS2000 DO-/Enter-Prozeduren, für CFS-Kommandoprozeduren (Kommando DO), sowie für Startup-Dateien von CFS.

Im Prozedurmodus von CFS werden die erforderlichen Eingaben (Selektionsbedingungen/ Variable Actions/Kommandos/Eingaben für Connections) von SYSDTA angefordert. Falls CFS Eingaben von SYSDTA entgegennehmen soll, muß der Prozeßschalter 1 auf ON gesetzt sein.

CFS kann in Prozeduren in zwei verschiedenen Betriebsarten genutzt werden:

a) **Basismodus:** In diesem Modus werden nur Eingaben akzeptiert, die der ersten CFS-Maske zugeordnet sind. Diese Eingaben können sein: Selektionsbedingungen, Variable Actions und Angaben für eine Dokumentationsdatei.

Der Programmablauf geht folgendermaßen von statten: Nachdem die gewünschten Datenobjekte aufgrund eines entsprechenden Eingabe-Datensatzes selektiert wurden, wird sofort zur Ausführung einer im Eingabe-Datensatz definierten Variablen Action (ON&...) übergegangen. Falls gewünscht wird eine Dokumentationsdatei der selektierten Datenobjekte erstellt.

Im Basismodus wird der gesamte Programmteil von CFS, der für die Ausführung von Kommandos zuständig ist (zweite CFS-Maske), übersprungen. In diesem prozedurgesteuerten Modus können mit relativ einfach aufgebauter Prozeduren, Datenobjekte selektiert und Variable Actions auf alle selektierten Datenobjekte angewendet werden.

Der Basismodus von CFS wird angenommen, wenn der Prozeßschalter 11 gesetzt ist. Außerdem müssen die Datensätze für die Dateiselektion mit dem Vorspann *001 beginnen. Näheres siehe Seite 487.

Der Basismodus ist nur möglich, falls CFS in BS2000 DO- oder Enter-Prozeduren aufgerufen wird. In CFS-Kommandoprozeduren ist der Basismodus nicht anwendbar.

b) **Erweiterter Prozedurmodus:** In diesem Modus wird der vollständige Bildschirmdialog mit allen erforderlichen Eingaben und Prompts simuliert.

Im erweiterten Prozedurmodus von CFS können Dateien in der zweiten Maske prozedurgesteuert mit individuellen Action-Codes versehen und entsprechend bearbeitet werden (z.B. Modify). Auch können in diesem Modus Connections eröffnet und Eingaben für die Connections getätigt werden. Letzteres sogar, falls CFS in einem Enter-Prozeß abläuft. In diesem Fall emuliert CFS ein 8-bit fähiges Terminal vom Typ 9763.

Im erweiterten Prozedurmodus müssen andererseits alle zum Ablaufzeitpunkt erforderlichen Programmeingaben, die sich oft erst aufgrund einer bestimmten Konstellation ergeben, exakt vorhergesehen und in der Prozedurdatei entsprechend enthalten sein.

Für den erweiterten Prozedurmodus darf Prozeßschalter 11 nicht gesetzt sein. Die Angabe zur Dateiselektion muß mit dem Vorspann *002 versehen sein. Evtl. folgende Eingaben für das Kommandofeld von CFS haben keinen Vorspann.

Wird CFS in einer BS2000 DO-Prozedur mehrmals hintereinander aufgerufen, so sollte vor dem zweiten und jedem weiteren Aufruf das Kommando
/SYSFILE SYSDTA=(SYSCMD) wiederholt werden.

Für den Prozedurmodus relevante Prozeßschalter

`/SETSW ON=1` CFS fordert alle benötigten Eingaben mit RDATA vom Eingabemedium SYSDTA an. In einer BS2000-Prozedur muß dazu das Kommando /SYSFILE SYSDTA=(SYSCMD) abgesetzt werden. /SETSW ON=1 wird sowohl bei bei Prozeduren mit *001-Sätzen (Basismodus) als auch bei Prozeduren mit *002-Sätzen (erweiterter Modus) benötigt.

`/SETSW ON=4` Der Umfang an Protokollierung und der Ausgaben von CFS wird auf ein Minimum reduziert. Im erweiterten Prozedurmodus werden insbesondere die Bildschirmmasken, die durch einen von SYSDTA gelesenen Satz mit Eingaben versorgt werden, nicht ausgegeben (WROUT).

`/SETSW ON=11` Voraussetzung für Prozeduren im Basismodus (*001).
Mit dem Schalter 11 muß stets auch der Schalter 1 gesetzt sein.
Der Schalter darf nicht gesetzt sein bei Prozeduren im erweiterten Modus (*002).

`/SETSW ON|OFF=31` Treten bei der Ausführung der Variablen Actions ON&COPY... / ON&ADD... / ON&SEL... Fehler auf, so wird der Prozeßschalter 31 gesetzt (/SETSW ON=31). Damit ist es für nachfolgende Prozedurteile möglich, die korrekte Ausführung der Variablen Action für alle beteiligten Dateien nachzuprüfen. Z.B. kann nach der Aufnahme von Dateien in eine Bibliothek ein automatischer ERASE-Lauf nur dann durchgeführt werden, wenn die korrekte Aufnahme aller Dateien sichergestellt ist (Prozeßschalter 31 ist nicht gesetzt).

Prozeßschalter 31 wird von CFS auch dann gesetzt, wenn keine Datenobjekte mit den gewünschten Auswahlkriterien gefunden bzw. wenn bei den Kommandos SI/ SIL/SIS keine Termine gefunden wurden.

Basismodus

Im Basismodus von CFS werden von SYSDTA nur Eingabesätze mit dem Vorspann *001 (eventuell mit Fortsetzung, siehe Beispiele) entgegengenommen.

Nach dem Einlesen der Selektionsbedingungen und der Variablen Actions werden von CFS keine weiteren Eingaben verlangt. Der Programmteil, der die Dateienliste am Bildschirm anzeigt und Kommandos bearbeitet, wird in diesem Fall übersprungen. CFS geht sofort zur Ausführung einer in der Form ON&... evtl. definierten Variablen Action über.

Nach Ausführung der Variablen Action für alle selektierten Datenobjekte wird auch die Terminierungsabfrage übersprungen. Falls im Eingabemedium SYSDTA ein neuer *001-Datensatz folgt, so wird dieser, wie oben beschrieben, als Eingabe für die Selektionsmaske von CFS betrachtet. CFS wird beendet, sobald die End of File-Bedingung für SYSDTA erreicht ist. Die End of File-Bedingung für SYSDTA tritt z.B. ein, wenn innerhalb einer DO-Prozedur als nächstes ein BS2000-Kommando folgt.

Die Eingaben für die Selektionsmaske von CFS werden im Basis-Prozedurmodus von CFS durch Datensätze mit dem Kennzeichen (Vorspann) *001 versorgt. Der Vorspann *001, zusammen mit den gesetzten Prozeßschaltern 1 und 11 bewirkt in CFS den Übergang in den Basis-Prozedurmodus.

Format der *001-Sätze

***001** *select*=*par* [; *select*=*par* .... ; *select*=*par*]

select Schlüsselwort eines Feldes der Selektionsmaske oder Abkürzung des Schlüsselwortes. Für nicht angegebene Schlüsselworte werden von CFS Spaces als Parameterwerte eingesetzt.

par Eingabewert für das entsprechende Feld der Selektionsmaske.

Als Trennungssymbol am Ende eines Parameterwertes vor dem Beginn eines neuen Schlüsselworts ist das Semikolon ";" zu verwenden.
Eingabesätze werden nur bis zu einer Länge von 80 Bytes ausgewertet. Falls ein Satz länger ist bzw. falls er aus Gründen der Übersichtlichkeit getrennt werden soll, so ist als Fortsetzungszeichen "-" an der letzten Stelle des Satzes anzugeben. Es sind beliebig viele Fortsetzungszeilen möglich. Unabhängig von der Möglichkeit der Fortsetzung ist zu beachten, daß jeder einzelne Parameter nur in der maximalen Länge des Eingabefeldes der Selektionsmaske angegeben werden kann. Für den Parameter FILENAME-SELECT sind dies z.B. 58 Byte.

Für die zu verwendenden Schlüsselworte gilt folgende Regel:

Grundsätzlich sind als Schlüsselworte die in der CFS-Selektionsmaske enthaltenen Feldbezeichnungen zu verwenden. Sie können jedoch verkürzt angegeben werden. Rechtsstehende Teile der Feldbezeichnung können bei der Angabe des Schlüsselworts weggelassen werden, solange eine eindeutige Zuordnung des verbleibenden Teils innerhalb aller Feldbezeichnungen der Selektionsmaske gewährleistet ist.

Anstelle von FILENAME-SELECT= würde es z.B. ausreichen, FI= anzugeben. F= wäre nicht mehr eindeutig, da zwei Felder in der Selektionsmaske mit dem Buchstaben F beginnen: FILENAME-SELECT und FCBTYPE.

Im folgenden sind die kürzestmöglichen Zuordnungen von Schlüsselworten zu den entsprechenden Feldern der Selektionsmaske von CFS aufgeführt:

FI	-->	FILENAME-SELECT
USER-I	-->	USER-ID
LA	-->	LAST PAGE
AG	-->	AGE
FCB	-->	FCBTYPE
VOL	-->	VOLUME
SH	-->	SHARE / NONSHARE
ACC	-->	ACCESS
PAS	-->	PASS
SO	-->	SORT OPTION
U	-->	USER OPTION
DOC	-->	DOCUMENTATION
VAR	-->	VARIABLE ACTION

Beispiele:

```
*001 FILENAME-SELECT=CFS;VARIABLE ACTION=ON&COPY ''='$TSOS.'

*001 USER-ID=(SYSDTA);USER OPTION=LACC>90;-
DOC=LST.EXPORT;VAR=ON&TSOSMT GEN SAVTAP,VSNOUT=(TAP001,TAP002)

*001 FILE=OLDLIST LISTE.EXPORT;VAR=ON&/ERASE

*001 FILE=.LST;AGE=0;VAR=ON&ADD BIBL.LST
```

Hinweise:

*001-Sätze dürfen in Startup-Dateien und CFS-Kommandoprozeduren nicht verwendet werden. In Prozeduren dieser Art sind für die Parameterversorgung der Selektionsmaske ausschließlich *002-Sätze zu verwenden (erweiterter Prozedurmodus, siehe unten).

Falls in einer CFS-Prozedur im Basismodus (*001), Dateien durch die Variable Action ON&ADD in eine noch nicht eingerichtete Bibliothek aufgenommen werden sollen (siehe letztes Beispiel), so erzeugt CFS eine Bibliothek vom Typ PLAM.

Parametrisierung durch Job- oder SDF-P Variable

Wird an irgendeiner Stelle in einem Datensatz für eine CFS-Prozedur das Konstrukt &(*name*) angegeben, so wird dieses vom CFS-Eingabeprozessor durch den momentanen Inhalt der Job- / SDF-P Variablen ersetzt. Hierbei kann für *name* auch eine CFS-Variable angegeben werden Die CFS-Variable muß als Wert den Namen der gewünschten Jobvariablen enthalten: &(&CFSVAR) .

Beispiele:

```
*001 FILE=&(CFS.SEL);SHARE=N;VAR=ON&/CAT RDPASS=&(PASSW),STATE=U

*&VAR1 = '&(&JVNAME)'
```

Erweiterter Prozedurmodus

Der erweiterte Prozedurmodus gestattet es, jegliche Art von Dateibearbeitung, sowie auch den Aufbau von Connections prozedurgesteuert durchzuführen. In diesem Modus ist dem Benutzer der volle Funktionsumfang von CFS auch in Prozeduren zugänglich. Dazu werden alle WRTRD-Aufrufe (Bildschirm-Eingabeaufforderungen) abgefangen und die Eingaben durch Datensätze von SYSDTA befriedigt. Die Nachrichtenköpfe bei Maskeneingaben werden von CFS selbst versorgt.

CFS verhält sich im erweiterten Prozedurmodus exakt in der gleichen Weise, wie dies auch im Dialog mit dem Benutzer am Bildschirm der Fall wäre. Der Ersteller einer derartigen Prozedur muß den Programmablauf daher möglichst eindeutig vorausbestimmen bzw. vorhersehen. Unvorhergesehene CFS-Anfragen, z.B. ob eine bestimmte Datei auf der Platte überschrieben werden soll oder nicht, würden in diesem Fall den ordnungsgemäßen Ablauf der Prozedur beeinträchtigen. Mit den Kommandos NQ*x* sind eventuelle Anfragen von vornherein zu unterbinden.

Für den erweiterten Prozedurmodus von CFS darf der Prozeßschalter 11 nicht gesetzt sein. Im Unterschied zum Basismodus sind die Eingaben für die Datenselektion mit *002 zu beginnen. Damit wird CFS darauf vorbereitet, daß nachfolgend Kommandos und andere Masken-Eingaben von SYSDTA gelesen werden. Ansonsten hat der *002-Satz das gleiche Format wie der *001-Satz.

Format der *002-Sätze

***002** *select*=*par* [; *select*=*par* .... ; *select*=*par*]

select Schlüsselwort eines Feldes der Selektionsmaske oder Abkürzung des Schlüsselwortes. Für nicht angegebene Schlüsselworte werden von CFS Spaces als Parameterwerte eingesetzt.

par Eingabewert für das entsprechende Feld der Selektionsmaske.

Als Trennungssymbol am Ende eines Parameterwertes vor dem Beginn eines neuen Schlüsselworts ist das Semikolon ";" zu verwenden.
Eingabesätze werden nur bis zu einer Länge von 80 Bytes ausgewertet. Falls ein Satz länger ist bzw. falls er aus Gründen der Übersichtlichkeit getrennt werden soll, so ist als Fortsetzungszeichen "-" an der letzten Stelle des Satzes anzugeben. Es sind beliebig viele Fortsetzungszeilen möglich. Unabhängig von der Möglichkeit der Fortsetzung ist zu beachten, daß jeder einzelne Parameter nur in der maximalen Länge des Eingabefeldes der Selektionsmaske angegeben werden kann. Für den Parameter FILENAME-SELECT sind dies z.B. 58 Byte.

Alle auf einen *002-Satz folgenden Eingaben (Kommandos und Eingaben für Connections) sind in der Prozedur direkt, d.h. ohne Vorspann anzugeben. Der nächste Satz, der die Daten für die Selektionsmaske von CFS zur Verfügung stellt, ist mit dem Vorspann *002 zu versehen.

Hinweise:

In einem CFS-Aufruf dürfen *001- und *002-Sätze nicht zusammen angegeben werden. Ein gemischter Betrieb zwischen Basis- und erweitertem Prozedurmodus ist nicht möglich.

Wird an irgendeiner Stelle in einem Datensatz einer CFS-Prozedur das Konstrukt &(*name*) angegeben, so wird dieses vom CFS-Eingabeprozessor durch den momentanen Inhalt der Job- / SDF-P Variablen ersetzt.

Prozedurvariable

Im erweiterten Prozedurmodus von CFS können Variable sowohl im Text für die Programmeingaben, als auch in den prozedurinternen Sprachelementen (*-Anweisungen) verwendet werden. Diese Variablen sind nicht zu verwechseln mit BS2000-Prozedurvariablen und können nur innerhalb von CFS-Prozeduren und Startup-Dateien verwendet werden.

Der Name einer CFS-Prozedurvariablen beginnt mit einem Einleitungszeichen, das standardmäßig auf '&' gesetzt ist. Mit der Prozeduranweisung *SET PAR=*x* (siehe nächster Abschnitt "Elemente der Prozedursprache") kann das Beginnzeichen für Parameter auf jedes andere Sonderzeichen umdefiniert werden. Der Name einer CFS-Prozedurvariablen kann 1 bis 20 alphanumerische Zeichen umfassen. Als einziges Sonderzeichen ist '-' innerhalb des Namens einer CFS-Prozedurvariablen erlaubt. Die Bezeichnung einer CFS-Prozedurvariablen kann auch mit einer Ziffer beginnen.

Der Inhalt von CFS-Prozedurvariablen kann 1 bis 80 Stellen umfassen. Beim Start der Prozedur bzw. bei Zuweisung des Nullstrings (&*var*='') ist der Inhalt der Variablen auf binär Null gesetzt. Ausnahme: CFS-interne Prozedurvariable. Diese Prozedurvariablen besitzen fest vorgegebene Namen und werden von CFS automatisch bei jeder Bezugnahme mit einem ihrer Bedeutung entsprechenden Inhalt versorgt. CFS-interne Prozedurvariable dürfen in der *PROC-Anweisung für CFS-Prozeduren (siehe Seite **512**) nicht auftreten.

Zweistufige Substitution: Der Inhalt einer Prozedurvariablen wird als Name einer anderen Prozedurvariablen interpretiert, falls die Variable in der Form &&*var* angegeben wurde. Beispiel: Der Variablen &CX wurde der String 'VAR1' zugewiesen. Durch die Angabe &&CX wird der Inhalt der Variablen &VAR1 angesprochen. Ein Beispiel für zweistufige Variablensubstitution finden Sie auf Seite **539**.

CFS-interne Prozedurvariablen

&ACCT	Abrechnungsnummer (ein bis acht Stellen) der Benutzerkennung, unter der CFS geladen wurde.
&ACCT(*tsn*)	Abrechnungsnummer des LOGON-Kommandos der angegebenen TSN. Die TSN muß stets vierstellig angegeben werden. Die Abrechnungsnummer einer fremden TSN kann nur ermittelt werden, falls CFS unter der Benutzerkennung TSOS geladen wurde. Für ein Beispiel dieser Form der Variablenverwendung siehe Seite **535**.
&ARGC	Anzahl der beim Aufruf an die CFS-Prozedur übergebenen Stellungsparameter. Leerparameter (par1,,par3) werden hierbei mitgezählt. Wurden keine Prozedurparameter übergeben, hat die Prozedurvariable den Wert 0.
&BS2PL	BS2000-Prozedurebene (Procedure Level Number). '0': CFS wurde nicht in einer DO-Prozedur aufgerufen. >'0': CFS wurde in einer DO-Prozedur des BS2000 aufgerufen.
&CFSINCON	wurde das aktuelle Programm CFS in einer Connection aufgerufen ? Y | N.

&CFSMSG — über diese bis zu 80 Stellen lange Variable kann auf die Nachricht zugegriffen werden, die bei nicht prozedurgesteuertem Ablauf von CFS in der Meldungszeile der letzten Bildschirmmaske ausgegeben worden wäre. Es werden sowohl die Fehlermeldungen aus der Dateienliste, als auch die Fehlermeldungen aus der Selektionsmaske an die Variable übergeben.

Die Variable CFSMSG hat den Leerstring ('') als Wert, falls die in der letzten Maske ausgegebene Meldung den Standardtext "Help: ?/?all/??" zum Inhalt hatte.

Beim Füllen der Variablen CFSMSG wird der Text der Nachricht in Großbuchstaben umgesetzt.

&CFSCOMP — über diese Variable kann die CFS-Maske abgefragt werden, in der die nächste Eingabe erfolgt. Die Variable kann die folgenden Werte annehmen:

'SEL'	Die nächste Eingabe erfolgt in der Selektionsmaske von CFS.
'CMD'	Die nächste Eingabe erfolgt im Kommandofeld der (eventuell auch leeren) Dateienliste.
'DISP'	Die nächste Eingabe erfolgt im Kommandofeld des CFS-Display/ Editors.
'CON*x*'	Die nächste Eingabe erfolgt in der CFS-Connection der Nummer *n*.

&CMDMSG — über diese bis zu 80 Stellen lange Variable kann auf die Meldung zugegriffen werden, die bei der zuletzt ausgeführten Prozeduranweisung Kommando */*cmd* (beliebiges BS2000-Kommando) ausgegebene wurde. Es werden sowohl die Fehlermeldungen, als auch Informationen des BS2000-Kommandos an die Variable übergeben.

Falls das BS2000-Kommando mehrere Meldungen ausgibt, so enthält die Variable CMDMSG nur den ersten Satz.

Die Variable CMDMSG hat den Leerstring ('') als Wert, falls die letzte Anweisung */*cmd* keine Meldung ausgegeben hat.

&COMPONENT — In dieser Variablen wird die Komponente angezeigt, in der sich der Benutzer bzw. die Prozedur momentan befindet: 'CFS' | 'TAS' | 'JES' | 'JESG'

&CON*x* — Ist eine Connection mit der angegebenen Nummer (0 <= *x* <= 9) eröffnet, so erhält die Variable als Wert den Namen der Anwendung, zu der die Connection eröffnet wurde (z.B. $DIALOG, $CONSOLE, ...). Ist keine Connection eröffnet, erhält die Variable als Wert den Leerstring ('').

&CON*x*H — Ist eine Connection mit der angegebenen Nummer (0 <= *x* <= 9) eröffnet, so erhält die Variable als Wert den Namen des Host-Rechners, zu dem die Verbindung besteht.

&CON*x*J — Ist eine Connection mit der angegebenen Nummer (0 <= *x* <= 9) zu $DIALOG eröffnet, so erhält die Variable als Wert den Jobnamen des Dialogprozeßes (.*name* LOGON).
Die Variable hat als Wert den Leerstring (''), falls
a) es sich nicht um eine Connection zu $DIALOG handelt,
b) im LOGON-Kommando kein Jobname angegeben wurde,
c) keine Connection mit der angegebenen Nummer eröffnet ist.

&CON*x*T — Ist eine Connection mit der angegebenen Nummer (0 <= *x* <= 9) zu $DIALOG eröffnet, so erhält die Variable als Wert die TSN des Dialogprozeßes. In allen anderen Fällen erhält die Variable als Wert den Leerstring ('').

&CON*x*U — Ist eine Connection mit der angegebenen Nummer (0 <= x <= 9) zu $DIALOG eröffnet, so erhält die Variable als Wert die Benutzerkennung des LOGON-Kommandos. In allen anderen Fällen erhält die Variable als Wert den Leerstring ('').

&DATE — aktuelles Tagesdatum in der Form: yy-mm-dd
&DATE4 — aktuelles Tagesdatum in der Form: yyyy-mm-dd (4-stellige Jahreszahl)
&DATE2 — aktuelles Tagesdatum in der Form: dd.mm.yy
&DATE42 — aktuelles Tagesdatum in der Form: dd.mm.yyyy (4-stellige Jahreszahl)

&DAY — Wochentag des aktuellen Datums. Der Wochentag ist in folgender Form codiert:
1 = Montag, 2 = Dienstag, 3 = Mittwoch, ..., 6 = Samstag, 7 = Sonntag.
&DAY2 — Wochentag des aktuellen Datums im Langtext in englisch.
&DAY2G — Wochentag des aktuellen Datums im Langtext in deutsch.

&DISP-POS — Gibt im Display-Modus bei SAM- und PAM-Dateien die Satz-/Blocknummer des ersten angezeigten Satzes. Bei ISAM-Dateien wird der Schlüssel des ersten angezeigten Satzes ausgegeben.

&DISPEOF — Zeigt an, ob im Display-Modus aufgrund des letzten Kommandos (z.B. +*n* oder S) das Ende der Datei erreicht wurde oder ob ein DMS-Fehler aufgetreten ist.
'N' Das letzte Kommando führte nicht auf die End-of-File Bedingung.
'Y' Das letzte Kommando führte auf die End-of-File Bedingung bzw. die Jobvariable konnte nicht gelesen werden.
'DMSnnnn' Es ist ein DVS-Fehler mit dem angegebenen Code aufgetreten.

&FROM — Wurde die CFS-Prozedur mit dem Fluchtsymbol aus einer Connection gestartet (^DO xxxx), so enthält die Variable &FROM die Nummer der Connection, aus der Prozeduraufruf erfolgte. Erfolgte der Prozeduraufruf in der Grundebene von CFS (DO xxxx), so enthält die Variable den Wert 'M' (Maintask).

&HOST — Bezeichnung des Host-Rechners (ein bis acht Stellen), an dem CFS geladen wurde.

&LISTEND — Die Variable enthält die Anzahl der Einträge der Dateienliste. Verborgene Einträge der Dateienliste werden hierbei nicht mitgezählt. Der Inhalt der Variablen ist leer (&LISTEND = ''), falls keine Dateienliste ausgewählt ist (FILENAME-SELECT=NO).
&LISTEND2 — Wie &LISTEND, jedoch werden verborgene Einträge der Dateienliste mitgezählt.

&LISTACT — Die Variable enthält den Action-Code der zu dem ersten in der Bildschirmmaske angezeigten Eintrag der Dateienliste gehört. Der Inhalt der Variablen ist leer (&LISTACT =''), falls kein Action-Code bei der ersten in der Maske angezeigten Datei eingetragen ist.

&LISTENTRY — Die Variable enthält den Inhalt des ersten in der aktuellen Bildschirmseite angezeigten Eintrags der Dateienliste. Der Inhalt der Variablen umfaßt 64 oder 69 Zeichen, je nachdem ob bei der Dateiselektion eine User Option angegeben wurde oder nicht.
&LISTENTRY2 — Wie &LISTENTRY, jedoch werden Datumsangaben (DATE-Option) und binäre Felder in Archive-Listeinträgen nicht in der CFS-internen Form, sondern wie am Bildschirm dargestellt.

&MONTH — Monat des aktuellen Datums in numerischer Form (z.B. 01 für Januar).
&MONTH2 — Monat des aktuellen Datums im Langtext in englisch.
&MONTH2G — Monat des aktuellen Datums im Langtext in deutsch.

&NEXTENTRY — Die Variable enthält den Inhalt des auf &LISTENTRY folgenden Eintrags der Dateienliste. Falls kein folgender Eintrag existiert, enthält die Variable den Wert '' (Nullstring).
&NEXTENTRY2 — Wie &NEXTENTRY, jedoch werden Datumsangaben (DATE-Option) und binäre Felder in Archive-Listeinträgen nicht in der CFS-internen Form, sondern wie am Bildschirm dargestellt.

&LOGNAM — Job-Name aus dem LOGON-Kommando (/.name LOGON ...).
Falls beim LOGON-Kommando kein Jobname angegeben wurde, so hat diese Variable als Wert den Leerstring ('').

&LOGNAM(*tsn*) — Job-Name des LOGON-Kommandos der angegebenen TSN. Die TSN muß stets vierstellig angegeben werden. Der Job-Name einer fremden TSN kann nur ermittelt werden, falls CFS unter der Benutzerkennung TSOS geladen wurde. Für ein Beispiel dieser Form der Variablenverwendung siehe Seite **535**.

&MODE — Prozeßmodus, in dem CFS abläuft: 'DIALOG' | 'BATCH'

&NEXTCON — enthält die Nummer einer noch nicht eröffneten Connection. Falls keine Connections eröffnet sind, hat NEXTCON den Wert 0. Falls Connection 0 bereits eröffnet ist, hat NEXTCON den Wert 1, usw. Falls alle Connections belegt sind, hat NEXTCON den Leerstring ('') als Wert.

&POSLIST — enthält eine fünfstellige Zahl, die die Position des ersten in der aktuellen Bildschirmseite angezeigten Eintrags der Dateienliste beschreibt. Siehe hierzu auch Kommando P*n* auf Seite **215**.

&POSUACT — enthält eine fünfstellige Zahl, die die Position des mit einem %-Action Code markierten Eintrags der Dateienliste beschreibt. Siehe hierzu auch Kommando P*n* auf Seite **215**.

&RDMASKIN — enthält die Bezeichnung der Taste, mit der die letzte mit *RDMASK ausgegebene Benutzermaske abgesendet wurde. Beispiel: DUE. DUE steht hier für die ENTER-Taste. Für weitere Informationen zu der Prozeduranweisung RDMASK siehe Seite **515**.

&SEARCHRET — Diese Variable enthält den Returncode des letzten Suche-Kommandos in CFS.

Returncode 'ERR'	Syntaxfehler bei der Definition des Suchbegriffs.
Returncode ''	Suchargument wurde nicht gefunden.
Returncode '*nnn*'	ein- bis sechsstellige Dezimalzahl. Bei reinem Suchen: Spalte des ersten Treffers Bei Suchen mit Ersetzen: Anzahl der gefundenen Treffer

&SEL-USID — PVS und User-ID der selektierten Datenobjekte.

&SLABEL — Name einer variablen Sprungmarke (*SKIP &SLABEL). Die Variable wird durch eine vorhergehende Anweisung *ON %OUT=(*datei*) versorgt. Die Datei, in der die Suchstrings der %OUT-Anweisung definiert sind, muß folgendes Format besitzen:
suchstring-1 .label-1
...
suchstring-n .label-n
&SLABEL enthält den Namen der Sprungmarke, die dem als erstes gefundenen Suchstring zugeordnet ist. Ein Anwendungsbeispiel hierzu finden Sie auf Seite **535**.

&SEL-LIB — Name der Bibliothek, in der die ausgewählten Elemente enthalten sind.

&STATNAM — Stations-Name (Name der Datensichtstation), an der CFS aufgerufen wurde.

&STATNAM(*tsn*) — Name der Datensichtstation, an der ein Dialogprozeß mit der angegebenen TSN eröffnet wurde. Die TSN muß stets vierstellig angegeben werden. Der Stationsname einer fremden TSN kann nur ermittelt werden, falls CFS unter der Benutzerkennung TSOS geladen wurde. Für ein Beispiel dieser Form der Variablenverwendung siehe Seite **535**.

&TIME — aktuelle Uhrzeit in der Form: hhmmss

&TIME2 — aktuelle Uhrzeit in der Form: hh:mm:ss

&TSN — vierstellige TSN des Benutzerprozeßes, unter dem CFS aufgerufen wurde.

&TEMP-PREF interner Prefix für Namen von Temporärdateien (S.*nnn*.) *nnn* = aktuelle Session-Number. Falls keine Temporärdateien aktiviert sind, bleibt die Variable leer.

&USER-ID Benutzerkennung (ein bis acht Stellen), unter der CFS geladen wurde.

&USER-ID(*tsn*) Benutzerkennung des LOGON-Kommandos der angegebenen TSN. Die TSN muß stets vierstellig angegeben werden. Die Benutzerkennung einer fremden TSN kann nur ermittelt werden, falls CFS unter TSOS geladen wurde. Für ein Beispiel dieser Form der Variablenverwendung siehe Seite 535.

Hinweise:

Das Ende des Namens einer Prozedurvariablen wird durch ein beliebiges Sonderzeichen außer '-' markiert. Ein Blank kann z.B. das Ende des Namens einer Variablen bezeichnen. Falls eine Variable im Datensatz einer CFS-Prozedur (kein Stern in Spalte 1) einen Teil des Textes substituiert, so ist zu beachten, daß ein Punkt hinter der Variablen bei der Substitution entfernt wird. Jedes andere Sonderzeichen als Abschluß einer Prozedurvariablen wird nicht aus dem Text entfernt.
Angenommen z.B., die Prozedurvariablen &N und &LOGNAM enthalten die Werte '1' und 'MEIER'. Der Datensatz "OC&N/.&LOGNAM.&N LOGON ..." hat nach Substitution der Variablen den Inhalt: "OC1/.MEIER1 LOGON ...".

Mit der zweistufigen Substitution können CFS-interne Variablen ausgewertet werden, deren Bezeichnung erst zum Ablaufzeitpunkt genau feststeht.
Angenommen, eine CFS-Prozedur TEST wird durch ^DO TEST aus der Connection 2 aufgerufen. In der Prozedur wird die Variable &CX durch die Anweisung &CX = 'CON'&FROM.'H' mit dem Inhalt 'CON2H', d.h. mit der Bezeichnung der CFS-internen Prozedurvariablen &CON2H versorgt. Die Prozeduranweisung *WRITE '&CX = &&CX' bewirkt die Programmausgabe 'CON2H = HOST1'. HOST1 ist dabei der Name des Hostrechners, zu dem Connection 2 eröffnet wurde.

Die in einer CFS-Prozedur verwendeten eigenen Variablen können beim Aufruf der Prozedur mit Werten versorgt werden. Dies geschieht in der gleichen Weise wie bei DO-Prozeduren im BS2000 durch Angabe der Parameter im Anschluß an den Namen der aufzurufenden Prozedur: DO prozedur, *par1*, *par2*, ...

Die Werte der in einer CFS-Prozedur angesprochenen Variablen müssen beim Aufruf der Prozedur entsprechend der verwendeten PROC-Anweisung als Stellungsparameter (*wert*) oder Schlüsselwortparameter (*par*=*wert*) angegeben werden. Für weitere Informationen hierzu siehe Seite 512 und 530.

CFS-Prozedurvariable können auch in Menue-Dateien verwendet werden. Siehe hierzu Seite 382.

Prozedurvariable und ***PROC-Anweisung**: Prozedurvariable werden mit ihrem ersten Auftreten innerhalb der Prozedur automatisch kreiert. Die CFS-Prozedursprache bietet die Möglichkeit, eine *PROC-Anweisung zu Beginn einer Prozedur anzugeben, in der alle oder einige der in der Prozedur verwendeten Variablen deklariert sind. Mehr zur *PROC-Anweisung siehe Seite 512.

Anweisungen der Prozedursprache

Alle Sprachelemente der Prozedursprache von CFS beginnen mit einem Stern (*).

Sprungmarke festlegen

.label

Definiert ein Sprungziel für eine *SKIP-Anweisung. Die Bezeichnung des Labels kann bis zu 16 Stellen lang sein. In einem *.label*-Satz dürfen keine weiteren Prozeduranweisungen oder Kommandoeingaben enthalten sein.

Kommentar

***** ⊔ [*comment*] | ***C**_[*comment*]

Datensätze, die nur aus einem '*' bestehen bzw. mit einem '*' beginnen und danach mindestens ein Blank enthalten, werden von CFS als Kommentar überlesen. Sie können zur Dokumentation der Prozedur verwendet werden.

" ... "

Prozeduranweisungen können in der gleichen Zeile mit einem Kommentar versehen werden. Die Kommentare sind hierbei in Anführungszeichen einzuschließen.
Beispiel: *&PTR=&PTR+1   "Laufvariable um 1 erhoehen"

Hinweis:

Das Kommando * kann in Prozeduren nicht zur Beendigung des Programms CFS verwendet werden, da '*' als Kommentar überlesen wird. Die Programmbeendigung innerhalb einer Prozedur ist nur über das Kommando END oder 'T' bei der Terminierungsabfrage möglich.

BS2000-Kommando

/cmd* [*param*]

Es wird das BS2000-Kommando *cmd* ausgeführt. Die wahlweise angegebenen Parameter ergänzen das Kommando. Sowohl *cmd* als auch *param* können CFS-Variable enthalten.
In der Anweisung */*cmd* sind beliebig viele Fortsetzungszeilen möglich. Als Fortsetzungszeichen ist '-' anzugeben. Im Unterschied zu BS2000 DO-Prozeduren beginnt eine Fortsetzungszeile mit '*'.

Die Meldungen bzw. Fehlermeldungen des Kommandos können über die interne CFS-Variable CMDMSG abgefragt werden.

Beispiel:

```
*&NAME='$CFS.CFS'
*/FSTAT &NAME
*WRITE '>&CMDMSG<'
```

Hinweis:

Das Kommando * kann in Prozeduren nicht zur Beendigung des Programms CFS verwendet werden, da '*' als Kommentar überlesen wird. Die Programmbeendigung innerhalb einer Prozedur ist nur über das Kommando END oder 'T' bei der Terminierungsabfrage möglich.

Simulieren einer Datenübertragungstaste

***K1** | *K3 ... | *K14 | *F1 | ... | *F5 | *DUE1 | *DUE2

Im Prozedurablauf wird die Betätigung einer K-/F-/ENTER-Taste simuliert. DUE1 steht für die Taste ENTER. DUE2 steht für die Taste Shift ENTER. Die Taste K2 kann nicht simuliert werden. Bezüglich K2 wird auf die Prozeduranweisung *ON %TIME auf Seite **510** verwiesen.

Beispiel:

*K1 ==> Im Prozedurablauf wird die Betätigung der K1-Taste simuliert.

Zuweisung für Prozedurvariable

***&*var1* =** [&*var2* | [C|L|X]'*string*'] [.&*var3* | .[C|L|X]'*string*'] ...

Der Prozedurvariablen &*var1* wird der Wert des nach dem '='-Zeichen angegebenen Ausdrucks zugewiesen. Einer Prozedurvariablen kann auch ein aus zwei numerischen Konstanten oder Variablen gebildeter Rechenausdruck zugewiesen werden. Die Syntax einer derartigen Zuweisung ist weiter unten beschrieben.

&*var1* &*name* [([*s*] [, *l*])]
Name der Prozedurvariablen, der ein Wert zugewiesen werden soll.

s wahlweise Angabe der Anfangsspalte ab der der rechts des Gleichheitszeichens angegebene Wert in die Variable &*var1* eingetragen wird. (1 <= *s* < 80).
Falls der Teil der Variablen &*var1* links von der Spalte *s* noch keinen Inhalt besitzt, so wird dieser mit Blanks gefüllt.
Standard: *s* = 1.

l wahlweise Angabe einer Länge, in der die Zuweisung an die Variable &*var1* erfolgen soll. (1 <= *l* <= 80).
Standard: *l* = Länge des rechts des Gleichheitszeichens stehenden Ausdrucks.

&*var2* &*name* [([*s*] [, *l*])]
Wert einer CFS-Prozedurvariablen oder eines Teilstrings einer Variablen.

&*name* Name einer CFS-Prozedurvariablen oder %PTR[0].
%PTR bezieht sich auf die Stelle, an der bei einer vorausgegangenen Anweisung *IF %OUT=... oder *ON %OUT=... der Trefferstring gefunden wurde.
%PTR(1,*l*) bezeichnet die Zeichenfolge in der Länge *l* <u>hinter</u> dem bei %OUT gefundenen Suchstring. Für ein Beispiel zu der Anwendung von %PTR siehe Seite **504**.
%PTR0(1,*l*) zeigt auf das erste Byte des bei %OUT gefundenen Suchstrings bzw. auf das erste Byte des ersten einer Reihe von mit * verknüpften Suchstrings.

s wahlweise Angabe der Anfangsspalte für einen Teilstring aus der Prozedurvariablen. (-80 < *s* < 80).
%PTR(*s*,*l*) bezeichnet eine Zeichenfolge in der Länge *l*, die *s* Stellen <u>hinter</u> dem letzten Byte des letzten bei %OUT gefundenen Suchstrings beginnt.
%PTR(-*s*,*l*) bezeichnet eine Zeichenfolge in der Länge *l*, die *s* Stellen <u>vor</u> dem letzten Byte des letzten bei %OUT gefundenen Suchstrings beginnt.
Standard: *s* = 1.

Beispiel:
Zu suchender Text: `123abc---------xyz789`

```
*IF %OUT='abc'*'xyz'
*&VAR=%PTR(1,3)     -->  &VAR='789'
*&VAR=%PTR(-3,4)    -->  &VAR='-xyz'
*&VAR=%PTR0(1,4)    -->  &VAR='abc-'
*&VAR=%PTR0(-3,4)   -->  &VAR='123a'
```

l wahlweise Angabe einer Länge für den Teilstring aus der angegebenen Prozedurvariablen. (1 <= *l* <= 80).
Standard: *l* = Länge der Variablen.

[C|L|X]'*string*' konstante Zeichenfolge. C'*string*' ist gleichbedeutend mit '*string*'. Bei L'*string*' wird die Zeichenfolge *string* nicht in Großbuchstaben umgewandelt. Innerhalb der Zeichenfolge vorkommende Hochkommas müssen verdoppelt werden ('').
Wird der Variablen ein numerischer Wert zugewiesen, so kann die Angabe der Hochkommas entfallen: *&*var*=[-]*zahl*
Beispiel: *&N=1, *&ZAEHLER=-10

.&*var3* Es können mehrere Teilausdrücke der Art .&*var*|.'*string*' aneinandergefügt werden. Vor dem zweiten und allen weiteren Teilausdrücken muß ein Punkt (.) als Trennungszeichen angegeben werden.

Hinweise:

Als Prozedurvariable &*var1* kann auch der Name einer bisher noch nicht vereinbarten Variablen angegeben werden. Die genannte Variable wird dann kreiert und kann in allen folgenden Anweisungen verwendet werden.

Eine Prozedurvariable kann auch mit einem Datenausschnitt aus einer Display-Datei versorgt werden. Näheres hierzu siehe Kommandos W (Write) Seite 327 und S,..=W (Suchen mit Wegschreiben) Seite 324.

***&*var1* = [&*var2* | *wert*] *op* [&*var3* | *wert*]**

Der Prozedurvariablen &*var1* wird der Wert des nach dem Zeichen = angegebenen arithmetischen Ausdrucks zugewiesen.

&*var2* | &*var3* Namen von CFS-Prozedurvariablen.

wert numerische Konstante: [-] *zahl*.

op arithmetischer Verknüpfungsoperator für eine der vier Grundrechenarten:
+ | - | * | /

Hinweise:

Um komplexe Modifikationen an Prozedurvariablen vorzunehmen, die evtl. in Abhängigkeit von bestimmten Suchbedingungen stehen, kann die Prozeduranweisung *RUN verwendet werden. Näheres hierzu auf Seite 525.

Werden bei arithmetischen Zuweisungen Prozedurvariablen angegeben, so müssen die Variablen numerische Werte enthalten ([-]*zahl*). Eine Ausnahme bildet lediglich die CFS-interne Variable &DATE (yy-mm-dd). &DATE kann arithmetisch mit +/- und einer Zahl verknüpft werden. Das Ergebnis ist wieder ein Datum.

Bei der Division zweier Werte wird das Ergebnis abgerundet. Bei Division durch Null wird an die Variable &*var1* kein Ergebnis zurückgegeben (entspricht &*var1*='').

Beispiele:

`*&T1='Connection 1: Anwendung '`
Die angegebene Zeichenfolge wird der Variablen &T1 als Inhalt zugewiesen.

`*IF &CON1 = ''    &T1 = &T1.'  nicht eröffnet'`
Falls Connection 1 nicht eröffnet ist (Inhalt der Variablen &CON1 = ''), so wird dem bisherigen Inhalt der Variablen &T1 die Zeichenfolge ' nicht eröffnet' angehängt.

`*IF &CON1 -= ''    &T1 = &T1.&CON1`
Falls die interne CFS-Variable &CON1 nicht leer ist (''), d.h. falls die Connection 1 eröffnet ist, wird an den bisherigen Inhalt der Variablen &T1 der Inhalt der Variablen &CON1 angefügt.

```
*&DZ = &DATE(7,2).'.'.&DATE(4,2).'.'.&DATE(1,2).'     '
*&DZ = &DZ.&TIME(1,2).':'.&TIME(3,2).':'.&TIME(5,2)
```
Aus den CFS-internen Variablen &DATE (yy-mm-dd) und &TIME (hhmmss) wird eine neue Variable mit dem Namen &DZ gebildet, die aufgrund des obigen Ausdrucks folgenden Inhalt hat: 'dd.mm.yy   hh:mm:ss'. Das gleiche Ergebnis wird auch durch die Anweisung `*&DZ=&DATE2.'     '.&TIME2` erreicht.

`*&ZAEHL=&VAR1+1`
Der Variablen &ZAEHL wird der um 1 erhöhte Wert von &VAR1 zugewiesen.

`*&VAR1=&VAR1-&N`
Der Wert der Variablen &VAR1 wird um den Inhalt der Variablen &N vermindert.

`*&D=&DATE+1`
Die Variable &D erhält das morgige Datum als Wert (yy-mm-dd).
Für weitere Beispiele für Variablenzuweisungen siehe Seite **539**.

Umschalten der Eingabequelle auf SYSCMD

***CON** SYSCMD

Die *CON-Anweisung ist nur in Startup-Dateien sinnvoll und bewirkt, daß die nächsten Eingaben vom Eingabemedium SYSCMD gelesen werden. Es ist damit möglich, CFS in einer DO-Prozedur aufzurufen, wobei der normale CFS-Startup durchgeführt wird (kein /SETSW ON=(1,4) in der DO-Prozedur). Am Ende des Startups kann die Eingabequelle durch die Anweisung *CON SYSCMD an die DO-Prozedur übertragen werden. Dies bedeutet, daß nach den allgemeinen Startup-Kommandos noch weitere spezielle Anweisungen aus der Prozedur ausgeführt werden.

Wurde CFS nicht in einer DO-Prozedur aufgerufen (Procedure Level Number 0), so hat die Anweisung *CON SYSCMD die gleiche Wirkung wie die Anweisung *DIA (siehe unten).

Eine in der Startup-Datei definierte Endebehandlung (*ON %END CONTINUE) wird nach Benutzung der Anweisung *CON korrekt verarbeitet.

Umschalten auf Benutzereingabe

***DIA** [*cmd*]

Die *DIA-Anweisung bewirkt, daß die folgenden Eingaben wieder vom Benutzer (Tastatur) angefordert werden (Umschaltung auf Benutzerdialog). Das Eingabemedium wird an die Tastatur zurückgegeben.

cmd Die wahlweise Angabe von *cmd* bewirkt, daß vor der Umschaltung auf Benutzerdialog das angegebene Kommando ausgeführt wird. Für *cmd* ist jedes Kommando und BS2000-Kommando zugelassen.

In einem prozedurgesteuerten EDT-Dialog empfiehlt es sich, vor Umschalten auf Benutzereingabe, die K3-Taste zur Auslösung des Bildschirm-Refresh zu simulieren (Anweisung *K3).

Beispiele:

```
*DIA /SYSFILE SYSDTA=(PRIMARY)
*DIA ^
*DIA RES CFS.LOG.xxx
```
(Rückkehr aus einer Connection in die Grundebene) – zu `*DIA ^`.

Weiteren Ablauf der Prozedur beenden

***EXIT** Diese Anweisung beendet die Prozedur. Es werden keine weiteren Prozeduranweisungen entgegengenommen. *EXIT kann an jeder Stelle der Prozedur stehen und beendet den weiteren Ablauf der Prozedur.

Normalerweise wird eine Prozedur mit der Ausführung der im letzten Datensatz stehenden Anweisung beendet. Eine *DIA-Anweisung beendet die Prozedur nicht. Es wird hier lediglich die Eingabekontrolle an den Benutzer zurückgegeben. Mit dem Kommando RET bzw. über eine latente ON-Bedingung (siehe 505) kann die Prozedur wieder fortgesetzt werden.

Im Zusammenhang mit dem Beenden von CFS-Prozeduren sei auch auf die Anweisung *RET SYSCMD verwiesen.

Beispiel:

```
*IF &N=0 EXIT
```
Falls der Inhalt der Variablen den Wert 0 besitzt, wird der weitere Ablauf der Prozedur beendet.

Bedingungen prüfen

Im folgenden werden 2 Formate der *IF-Anweisung dargestellt. *IF bewirkt einen Sprung innerhalb der Prozedur, falls eine bestimmte Bedingung erfüllt ist.

***IF** &[&]*var1 vgl* [[C|L|X] '*string*' | *zahl* | &[&]*var2*] *cmd*

Der Inhalt der Prozedurvariablen &*var1* wird mit der in Hochkommas eingeschlossenen Zeichenfolge *string*/mit dem Inhalt der Prozedurvariablen &*var2* verglichen.

&*var1* | &*var2*

Für &*var1* bzw. &*var2* kann jeweils folgendes Konstrukt, bestehend aus Variablenname, Anfangsspalte und Länge angegeben werden.

&*var* [([*s*] [, *l*])]
Bezeichnung eines Teilstrings einer Variablen.

s Anfangsspalte für den Teilstring der Variablen. (1 <= *s* < 80).
Standard: *s* = 1 (Spalte 1).

l Länge des Teilstrings. (1 <= *l* <= 80).
Im ersten Vergleichsoperanden (&*var1*) können für *l* auch die Sonderzeichen * bzw. $ angegeben werden. Die besondere Bedeutung dieser Zeichen ist in einen nachfolgenden Abschnitt genau beschrieben.
Standard: *l* = Gesamtlänge der Variablen.

Beispiele:

`&DATEI(4)`	Inhalt der Variablen &DATEI ab Spalte 4 bis zum Ende
`&DATEI(4,20)`	Inhalt der Variablen &DATEI ab Spalte 4 in der Länge 20
`&DATEI(,20)`	Inhalt der Variablen &DATEI ab Spalte 1 in der Länge 20

&&*var1* | &&*var2*

Der Inhalt der Variablen &*var1* | &*var2* wird als der Name einer anderen Variablen betrachtet, deren Inhalt dann ausgewertet wird.
Beispiel: &NA='CON1'. Durch &&NA wird auf den Inhalt der CFS-internen Variablen &CON1 Bezug genommen. Ein weiteres Beispiel für zweistufige Variablensubstitution finden Sie auf Seite **539**.

vgl Vergleichsoperator: [-] { > | < | = }
Fällt der alphanumerische Vergleich auf größer (>) / kleiner (<) / gleich (=) positiv aus, bzw. nicht positiv bei vorangestelltem '-' , so wird das unter *cmd* angegebene Kommando ausgeführt. Anstelle von -= (ungleich) kann auch <> angegeben werden.

[C|L|X]'*string*' konstante Zeichenfolge. Innerhalb der Zeichenfolge vorkommende Hochkommas müssen verdoppelt werden ('').
Der Inhalt der links des Gleichheitszeichens stehenden Variablen bzw. deren Teilstring wird mit der angegebenen Konstanten verglichen. Gleichheit ist nur dann gegeben, wenn sowohl die entsprechenden Längenmerkmale, wie auch die Dateninhalte übereinstimmen. Überschüssige Leerzeichen im Variableninhalt, die in der Konstanten nicht enthalten sind, führen zu einem negativen Vergleichsergebnis. Näheres hierzu siehe nächsten Abschnitt.

zahl Falls der Vergleichswert *string* einen Zahlenwert beinhaltet, so sind zwei Fälle zu unterscheiden:

a) Angabe von *string* in Hochkommas '*zahl*': Der String wird vor dem Vergleich intern in eine sechsstellige Zahl mit führenden Nullen umgewandelt.
b) Bei Angabe von *zahl* ohne Hochkommas wird der String in der angegebenen Länge 1 zu 1 für den Vergleich übernommen.

Beispiel: `*&N='261197'`
`*IF &N(1,2) < 27 SKIP .LT` --> Vergleich ist positiv ('26' < '27')
`*IF &N(1,2) < '27' SKIP .LT` --> Vergleich ist negativ ('26' > '000027')

cmd beliebige Anweisung der Prozedursprache von CFS. Der normalerweise eine Prozeduranweisung einleitende Stern darf nicht angegeben werden.

Sonderbedeutung der Längenangaben * bzw. $ beim Vergleich mit einer Konstanten

* Vergleiche die linksstehende Variable nur in der Länge der rechts des Gleichheitszeichens angegebenen Konstanten. Ist die Variable länger, so werden diese Zeichen für den Vergleich nicht herangezogen. Die Angabe &VAR(,*) kann in der Form &VAR(*) abgekürzt werden.

$ Expandiere die rechts des Gleichheitszeichens angegebene Konstante durch Aneinanderfügen solange, bis sich eine Zeichenfolge ergibt, die die gleiche Länge aufweist wie der Inhalt der zu vergleichenden Variablen.
Die Angabe &VAR(,$) kann in der Form &VAR($) abgekürzt werden.

Beispiele:
`*IF &DATEI(,*)='ABC' SKIP .ABC` oder in vereinfachter Schreibweise:
`*IF &DATEI(*)='ABC' SKIP .ABC`
Der Anfang der Variablen &DATEI wird in der Länge 3 mit der Konstanten 'ABC' verglichen und ggf. ein Sprung zum Label .ABC ausgeführt.

`*IF &DATEI(25,*)=' ' SKIP .A25`
Der Inhalt der Variablen &DATEI wird ab Spalte 25 in der Länge 1 mit der Konstanten '␣' verglichen. Bei Gleichheit wird ein Sprung zum Label A25 durchgeführt.

`*IF &DATEI(,$)=' ' SKIP .A27` oder in vereinfachter Schreibweise:
`*IF &DATEI($)=' ' SKIP .A27`
Der gesamte Inhalt der Variablen &DATEI wird mit einer Konstanten mit entsprechend vielen Leerzeichen ' ' verglichen. Falls der Inhalt von &DATEI nur aus Leerzeichen besteht, wird ein Sprung zum Label .A27 ausgeführt.

`*IF &DATEI(25,$)=' ' SKIP .A25`
Es wird geprüft, ob die Variable &DATEI in Spalte 25 oder höher nur noch Leerzeichen (mindestens eines) enthält. Ist dies der Fall, so wird ein Sprung zum Label .A25 ausgeführt.

Hinweise:

Beim Vergleich einer Prozedurvariablen mit einer Konstanten, bzw. mit einer zweiten Prozedurvariablen wird Groß-/Kleinschreibung standardmäßig nicht berücksichtigt. Durch die Anweisung *SET CASE=Y kann die Berücksichtigung von Groß-/Kleinschreibung jedoch eingeschaltet werden.

Zum Prüfen, ob der Inhalt einer Variablen eine bestimmte Länge (z.B. 6 Zeichen) nicht überschreitet, ist folgende Abfrage zu verwenden:
`*IF &VSN(7,1) = X'00' SKIP .KUERZER-ALS-7`

Beispiele:

`*IF &DAT='' PROMPT &DAT,'Bitte Dateinamen eingeben '`
Falls die Variable &DAT noch keinen Wert zugewiesen hat, so wird über die Prozeduranweisung *PROMPT vom Benutzer ein Wert für die Variable angefordert.

`*IF &DAT=&D2 SKIP .GLEICH`
Falls die Variable &DAT in ihrem Wert mit der Variablen &D2 übereinstimmt, so wird ein Sprung zu dem Label ".GLEICH" durchgeführt.

`*IF &VAR1(2,3)=&D2(1,3) WRITE 'GLEICH'`
Falls der Inhalt der Variablen &VAR1 in Spalte 2, 3 und 4 mit dem Inhalt der Variablen &D2 in Spalte 1, 2 und 3 übereinstimmt, so wird der Text 'GLEICH' am Bildschirm ausgegeben.

`*IF &ZAEHLER>9 EXIT`
Der Ablauf der Prozedur wird beendet, falls die Variable &ZAEHLER einen Wert größer als 9 besitzt.

Weitere Beispiele für Variablenzuweisungen finden Sie auf Seite **539**.

***IF SETSW** [ON|OFF]=*n cmd*

Der Prozeßschalter *n* wird getestet. Stimmt die Stellung des Schalters mit dem angegebenen Wert überein, so wird das Kommando *cmd* ausgeführt.

cmd beliebige Anweisung der Prozedursprache von CFS. Der normalerweise vor einer Prozeduranweisung stehende Stern darf nicht angegeben werden.

Hinweise:

Der Prozeßschalter 31 wird von CFS als Fehlerindikator verwendet. Z.B. wird der Schalter gesetzt, falls bei der Selektion keine Datenobjekte mit den gesuchten Eigenschaften gefunden wurden. Im prozedurgesteuerten Modus wird in diesem Fall so verfahren, als ob im Feld FILENAME-SELECT : NO eingegeben worden wäre. Es wird also in die zweite CFS-Maske verzweigt, wobei lediglich das Feld COMMAND: angezeigt wird. Im nicht prozeduralen Modus erscheint die Selektionsmaske mit einem entsprechenden Fehlerhinweis.

Prozeßschalter 31 wird von CFS auch bei Fehlern während der Ausführung der Variablen Actions ONXCOPY/ONXSEL/ONXADD gesetzt.

Um für spätere Abfragen eine definierte Ausgangssituation herzustellen, sollte der Schalter 31 vor jeder Aktion von CFS, die diesen Schalter verändern könnte, zurückgesetzt werden (Kommando /SETSW OFF=31).

***IF %OUT** [(%PTR)] =*such cmd*

Die Bedingung *IF %OUT ist erfüllt, wenn der Suchstring in der letzten Ausgabenachricht (z.B. aufgrund eines BS2000-Kommandos) oder Bildschirmmaske enthalten ist. Ist dies der Fall, wird das unter *cmd* angegebene Kommando ausgeführt. Außerdem wird ein Pointer %PTR auf das Ende des gefundenen Suchstrings in der Ausgabenachricht gesetzt. Der Pointer kann dazu benutzt werden, einen Teilstring der Ausgabenachricht an eine Prozedurvariable zuzuweisen.

such einfaches oder mehrfaches Suchargument der Form
[C]'*string*' | L'*string*' | A'*string*' | X'*string*'
Es ist zu beachten, daß bei Angabe des Suchstrings in der Form C'*string*' Kleinbuchstaben in *string* in die entsprechenden Großbuchstaben umgewandelt werden.
Die Angabe '*string*' hat dieselbe Wirkung wie A'*string*' (siehe unten).
Bei Angabe von L'*string*' bleibt *string* in der angegebenen Groß-/Kleinschreibung unverändert.
Bei Angabe von A'*string*' (Any) ist die Schreibweise von *string* in Klein- oder Großbuchstaben für das Suchergebnis ohne Bedeutung. Sowohl der Suchstring, als auch der zu durchsuchende Text werden vor dem Vergleich in Großbuchstaben umgewandelt.

Hinweise:

Die *IF %OUT-Abfrage kann auch auf die Ausgabenachrichten eines vorausgegangenen */bs2cmd* (eingegeben im Kommandofeld von CFS) oder einer vorausgegangenen Prozeduranweisung **/bs2cmd* angewendet werden. Hierbei ist jedoch zu beachten, daß die Ausgaben des vorhergehenden BS2000-Kommandos nur mit einer *IF %OUT Anweisung abgefragt werden dürfen.

Bei einer vorausgehenden */*bs2cmd* Prozeduranweisung ist zudem folgende Besonderheit zu berücksichtigen: Mit der nächsten *IF %OUT-Abfrage wird auf die Ausgaben des Kommandos zugegriffen. Danach, d.h. mit der übernächsten und evtl. folgenden *IF-%OUT Abfragen wird wieder auf die letzte Maskenausgabe zugegriffen. Als Dummy-Abfrage könnte z.B. angegeben werden *IF %OUT='' &XXX='xxx'. Nach dieser Anweisung wird mit *IF %OUT wieder auf die Ausgabe der letzten Maske zugegriffen.

Mit der Anweisung *IF %OUT='' wird %PTR auf das Ende der letzten Ausgabenachricht positioniert. Mittels *&VAR=%PTR(-80,80) werden z.B. die letzten 80 Byte der Ausgabenachricht in die Variable &VAR übertragen.

Beispiele:

`*IF %OUT=A'search-item found and replaced' &COUNT=%PTR(1,7)`
Der Ausdruck (1,7) nach %PTR wird wie bei einer Prozedurvariablen als Offset und Länge für die Zuweisung interpretiert. Mit dem Offset 1 wird die Stelle direkt nach dem letzten Suchitem bezeichnet.

`*IF %OUT='ENTRIES IN THIS LIST' &COUNT=%PTR(-6,5)`
Mit dem Offset -6 wird die Stelle 6 Byte vor dem Suchitem bezeichnet.

(%PTR) Ausgehend von einer vorhergehenden Anweisung *IF %OUT wird in der Ausgabenachricht ab der Spalte, in der der letzte Suchbegriff gefunden wurde, nach dem in *such* angegebenen String gesucht.

such einfaches oder mehrfaches Suchargument der Form '*string*' | L'*string*' | X'*string*'. L'*string*' wirkt wie '*string*' mit dem Unterschied, daß eingegebene Kleinbuchstaben nicht in Großbuchstaben umgesetzt werden. Mehrere Suchargumente können durch die Operatoren + (Und) / , (Oder) / * (Wildcard) verknüpft werden.
Eine ausführliche Beschreibung der Suchbedingung finden Sie auf Seite **309** und **313**.

cmd Anweisung der Prozedursprache von CFS. Der normalerweise vor einer Prozeduranweisung stehende Stern darf nicht angegeben werden.

Beispiel:

Mit der folgenden Prozedur wird verhindert, daß der Benutzer mit der K1-Taste in die Selektionsmaske von CFS gelangt. Für eine Beschreibung der ON %OUT-Bedingung siehe Seite **509**.

```
.ON-IN
*ON %IN=K1 CONTINUE
*DIA
*IF %OUT='FILENAME-SELECT :' SKIP .INPUT-NO
*SKIP .ON-IN
.INPUT-NO
NO
*SKIP .ON-IN
```

Prozedurvariablen aus Datei laden

***LV** *datei* [, C|O]

datei Aus der angegebenen Datei werden die Namen und Werte von CFS-Prozedurvariablen gelesen und in der aktuellen Prozedur verfügbar gemacht. Die Variablen sind danach so ansprechbar, als wären sie zuvor mit einer Zuweisung *&*var*='*wert*' erzeugt und mit einem Inhalt versehen worden.

C — Clear. Alle bisher in der Prozedur erzeugten Variablen werden vor der Ausführung der *LV-Anweisung gelöscht. Danach stehen nur die in der LV-Datei gesicherten Variablen zur Verfügung.

O — Overwrite. Die bisher in der Prozedur erzeugten Variablen stehen auch nach der Ausführung der *LV-Anweisung zur Verfügung. Gleichnamige Variablen, die in der Prozedur vor der *LV-Anweisung erzeugt wurden und die auch in der LV-Datei enthalten sind, werden mit den in der LV-Datei gespeicherten Inhalten überschrieben.

Standard: Falls keine der Optionen C|O angegeben wurde, werden gleichnamige Variablen durch die in der LV-Datei gespeicherten Inhalte nicht überschrieben.

Ereignisse für latente Bedingung festlegen

Im folgenden werden 9 verschiedene Varianten der ON-Bedingung dargestellt.

Mit der Prozeduranweisung *DIA kann der Ablauf der Prozedur unterbrochen und die Kontrolle an den Benutzer übergeben werden. Die Prozedur kann in diesem Fall nur über das vom Benutzer einzugebende Kommando RET (siehe Seite 517) wieder aufgenommen werden.

ON-Anweisungen bieten ein Mittel, den Prozedurablauf bei bestimmten Bedingungen automatisch fortzusetzen. ON-Anweisungen müssen vor einer *DIA-Anweisung (Umschaltung auf Benutzereingabe) angegeben werden. Sie definieren latente Bedingungen. Bei jeder Benutzereingabe prüft CFS, ob eine ON-Bedingung erfüllt ist. Im positiven Fall wird der Prozedurablauf automatisch fortgesetzt.

Die Anweisung *RESET setzt eine zuvor definierte ON-Bedingung zurück. Die RESET-Anweisung ist notwendig, wenn eine ON-Bedingung zurückgesetzt werden soll, obwohl diese noch nicht eingetreten ist. Bei Eintritt, d.h. bei Wirksamwerden einer ON-Bedingung, wird diese automatisch zurückgesetzt. Eine ON-Bedingung einer bestimmten Kategorie (%LST, %CMD, %TERM, %END) kann aufgrund einer Definition höchstens einmal wirksam werden.

***ON %ABEND** CONTINUE | SKIP *.label*

*ON %ABEND ist erfüllt, wenn CFS abnormal beendet wird. Dies ist z.B. der Fall bei Ausfall des Vorrechners (Leitungsverlust) oder durch /EXEC, /LOAD, /DO, /CALL, /LOGOFF im Kommandomodus des BS2000 bei noch geladenem CFS. Wenn die %ABEND-Bedingung erfüllt ist, wird der prozedurgesteuerte Modus aufgenommen und die im Anweisungsteil festgelegte Aktion ausgeführt.

CONTINUE — Der prozedurgesteuerte Ablauf wird wieder aufgenommen an der Stelle, an der die Prozedur durch die letzte *DIA-Anweisung unterbrochen wurde.

SKIP *.label* — Der prozedurgesteuerte Ablauf wird wieder aufgenommen. Es wird innerhalb der Prozedur ein Sprung zu der angegebenen Sprungmarke durchgeführt. Anstelle von SKIP kann auch GOTO angegeben werden.

***ON %CMD** CONTINUE [AFTER INPUT] | SKIP [AFTER INPUT] *.label*

Nachdem der Benutzer von der Prozedur die Kontrolle erhalten hat, prüft CFS vor jedem Eingabeschritt, ob mit der als nächstes anstehenden Eingabe ein Kommando angefordert wird. Ist dies der Fall, so ist die Bedingung *ON %CMD erfüllt und es wird der Anweisungsteil der ON-Anweisung ausgeführt.

CONTINUE — Der prozedurgesteuerte Ablauf wird automatisch wieder aufgenommen vor dem ersten Eingabeschritt, der das Kommandofeld von CFS versorgt.

CONTINUE AFTER INPUT
Der prozedurgesteuerte Ablauf wird automatisch wieder aufgenommen nach der ersten Benutzereingabe, die das Kommandofeld von CFS versorgt (z.B. auch Absenden der leeren CFS-Maske).

SKIP *.label* Der prozedurgesteuerte Ablauf wird automatisch mit dem ersten Eingabeschritt aufgenommen, der das Kommandofeld von CFS versorgt. Es wird in der Prozedur ein Sprung zu der angegebenen Sprungmarke durchgeführt. Anstelle von SKIP kann auch GOTO angegeben werden. Anstelle von SKIP kann auch GOTO angegeben werden.

SKIP AFTER INPUT *.label*
Der prozedurgesteuerte Ablauf wird automatisch wieder aufgenommen nach der ersten Benutzereingabe, die unter anderem auch das Kommandofeld von CFS versorgt. Es wird in der Prozedur ein Sprung zu der angegebenen Sprungmarke durchgeführt.

Beispiel:

Die folgende Prozedur eröffnet eine Connection zu $CONSOLE. Nach Rückkehr aus der Connection in CFS wird diese automatisch wieder geschlossen (*ON %CMD CONTINUE).

```
OC9$CONSOLE
xxxxx,C'sssss'
*ON %CMD CONTINUE
*DIA
DC9
```

***ON %END** CONTINUE | SKIP *.label*

*ON %END ist erfüllt, wenn durch Benutzereingaben (Kommando */END, K1-Taste) die Beendigung des Programms CFS angefordert wird. Wenn die %END-Bedingung erfüllt ist, wird der prozedurgesteuerte Modus wieder aufgenommen und die im Anweisungsteil festgelegte Aktion wird ausgeführt.

CONTINUE Der prozedurgesteuerte Ablauf wird wieder aufgenommen an der Stelle, an der die Prozedur durch die letzte *DIA-Anweisung unterbrochen wurde.

SKIP *.label* Der prozedurgesteuerte Ablauf wird wieder aufgenommen. Es wird innerhalb der Prozedur ein Sprung zu der angegebenen Sprungmarke durchgeführt. Anstelle von SKIP kann auch GOTO angegeben werden.

Hinweise:

Die Bedingung *ON %END wird auch aktiv, falls CFS durch eines der Kommandos /LOGOFF, /DO, /CALL, /EXEC, /LOAD, quasi gewaltsam beendet wird. Die genannten Kommandos werden von CFS zurückgestellt, bis alle Kommandos der Enderoutine abgearbeitet sind. Danach wird das die Beendigung von CFS verursachende BS2000-Kommando ausgeführt.

Die Bedingung *ON %END ist nicht erfüllt, falls CFS abnormal beendet wird (siehe *ON %ABEND-Anweisung weiter unten). Die Bedingung *ON %END ist ebenfalls nicht erfüllt, falls CFS prozedurgesteuert durch das Kommando END oder die Eingabe T bei der Terminierungsabfrage beendet wird. Das Kommando END, angegeben innerhalb der *ON %END-Kommandofolge hat die bedingungslose Beendigung des Programms CFS zur Folge.

Mit der Anweisung *ON %END kann eine Kommandofolge zum Beenden von CFS aktiviert werden (CFS-Shutdown). Diese Kommandofolge ist das Gegenstück zum CFS-Startup. Es können hier z.B. Connections kontrolliert beendet werden. Des weiteren kann das Kommandogedächtnis in eine Datei bzw. in eine Bibliothek gesichert werden.

Ein Anwendungsbeispiel für die *ON %END-Anweisung findet sich auf Seite **537** und folgende (Startup).

***ON %IN**= { *t-bez* | *such* } CONTINUE | SKIP *.label*

t-bez Tastenbezeichnung: K1, K3, ... , K23 / F1, ... , F20 / DUE1 / DUE2

Diese Variante der %IN-Bedingung ist erfüllt, nachdem der Benutzer die bezeichnete Datenübertragungstaste betätigt hat. DUE1 steht hierbei für ENTER und DUE2 für Shift ENTER. Es wird der prozedurgesteuerte Modus aufgenommen (CONTINUE) bzw. ein Sprung zu dem angegebenen Label ausgeführt (SKIP *.label*).

Die *ON %IN Anweisung kann nicht verwendet werden, um bestimmte DUE-Tasten in *RDMASK abzufragen. Um die Datenübertragungstaste der letzten RDMASK Maske abzufragen ist nach *RDMASK die CFS-interne Variable &RDMASKIN auszuwerten. Dazu ist die folgende Anweisung anzugeben: ***IF &RDMASKIN** = *t-bez* SKIP *.label*
Beispiel: *IF &RDMASKIN = K1 SKIP .ENDE

such einfaches oder mehrfaches Suchargument der Form
'*string*' | L'*string*' | A'*string*' | X'*string*'
Es ist zu beachten, daß bei Angabe des Suchstrings in der Form '*string*' Kleinbuchstaben in *string* in die entsprechenden Großbuchstaben umgewandelt werden.
Bei Angabe von L'*string*' bleibt *string* in der angegebenen Groß-/Kleinschreibung unverändert. Es wird keine Umsetzung in Großbuchstaben durchgeführt.
Bei Angabe von A'*string*' (Any) ist die Schreibweise von *string* in Klein- oder Großbuchstaben für das Suchergebnis ohne Bedeutung. Sowohl der Suchstring, als auch der zu durchsuchende Text werden vor dem Vergleich in Großbuchstaben umgewandelt.
Mehrere Suchargumente können durch die Operatoren + (Und) / , (Oder) / * (Wildcard) verknüpft werden.

Die %IN-Bedingung ist erfüllt, wenn der Suchstring bzw. einer der Suchstrings in der vom Benutzer getätigten Eingabe enthalten ist. Es wird der prozedurgesteuerte Modus aufgenommen (CONTINUE) bzw. ein Sprung zu dem angegebenen Label in der Prozedur ausgeführt. Außerdem wird ein Pointer %PTR auf das Ende des gefundenen Suchstrings in der Eingabenachricht gesetzt. Der Pointer kann dazu benutzt werden, einen Teilstring der Eingabenachricht an eine Prozedurvariable zuzuweisen. Weitere Informationen hierzu siehe Seite **504**.

Eine ausführliche Beschreibung der Suchbedingung finden Sie auf den Seiten **309** und **313**.

***ON %IN** Die zuletzt definierte *ON %IN-Bedingung wird wieder aktiviert.

Ein Beispiel für die ON %IN-Anweisung finden Sie auf Seite **503**.

***ON %INP** CONTINUE

*ON %INP ist erfüllt nach der ersten Benutzereingabe nach einer Umschaltung auf Benutzerdialog durch eine *DIA-Anweisung. Wenn die %INP-Bedingung erfüllt ist, wird der prozedurgesteuerte Modus wieder aufgenommen (CONTINUE).

Hinweise:

Mit der Anweisung *ON %INP ist es möglich, daß der Benutzer während des Ablaufs der Prozedur (z.B. im Dialog mit EDT oder in einer Connection) manuell genau eine Eingabe tätigt und danach der Prozedurlauf automatisch fortgesetzt wird.

Nach der Anweisung *ON %INP können noch andere Prozeduranweisungen folgen. %INP wird nach *DIA (Umschalten auf Benutzereingabe) aktiv.

Beispiel:

```
*002 FILE=NO
EDT
READ'datei'
LOWER ON
*WRITESYS 'Bitte Änderungen vornehmen. Prozedur läuft danach automatisch
weiter'
*F2                     F2-Taste:  EDT-Bildschirm überschreibbar machen
*K3                     K3-Taste:  Bildschirm neu ausgeben
*ON %INP CONTINUE
*DIA
WRITE'datei'
HALT
END
```

***ON %LST** CONTINUE [AFTER INPUT] | SKIP [AFTER INPUT] *.label*

Nachdem der Benutzer von der Prozedur die Kontrolle erhalten hat, prüft CFS nach jedem Dialogschritt, ob aufgrund der getätigten Benutzereingabe, ein Übergang vom Display- in den List-Modus erfolgt (Übergang aus dem Display-Modus in die Anzeige der Dateienliste durch Betätigen der **K1**-Taste oder durch Eingabe des Kommandos LST). Ist dies der Fall, so ist die Bedingung *ON %LST erfüllt und es wird der Anweisungsteil der ON-Anweisung ausgeführt.

CONTINUE Der prozedurgesteuerte Ablauf wird vor dem ersten Kommando im List-Modus automatisch wieder aufgenommen.

CONTINUE AFTER INPUT
Der prozedurgesteuerte Ablauf wird nach dem ersten vom Benutzer eingegebenen Kommando im List-Modus automatisch wieder aufgenommen.

SKIP *.label* Der prozedurgesteuerte Ablauf wird vor dem ersten Kommando im List-Modus (Anzeige der Dateienliste) automatisch wieder aufgenommen. Es wird in der Prozedur ein Sprung zu der angegebenen Sprungmarke durchgeführt. Anstelle von SKIP kann auch GOTO angegeben werden.

SKIP AFTER INPUT *.label*
Der prozedurgesteuerte Ablauf wird nach dem ersten vom Benutzer eingegebenen Kommando im List-Modus automatisch wieder aufgenommen. Es wird in der Prozedur ein Sprung zu der angegebenen Sprungmarke durchgeführt.

***ON %OUT**=*such* CONTINUE [AFTER INPUT] | SKIP [AFTER INPUT] *.label* | RETURN | RESUME

such einfaches oder mehrfaches Suchargument der Form

[C]'*string*' | L'*string*' | A'*string*' | X'*string*'

Es ist zu beachten, daß bei Angabe des Suchstrings in der Form C'*string*' Kleinbuchstaben in *string* in die entsprechenden Großbuchstaben umgewandelt werden.

Die Angabe '*string*' hat dieselbe Wirkung wie A'*string*' (siehe unten).

Bei Angabe von L'*string*' bleibt *string* in der angegebenen Groß-/Kleinschreibung unverändert.

Bei Angabe von A'*string*' (Any) ist die Schreibweise von *string* in Klein- oder Großbuchstaben für das Suchergebnis ohne Bedeutung. Sowohl der Suchstring, als auch der zu durchsuchende Text werden vor dem Vergleich in Großbuchstaben umgewandelt.

Mehrere Suchargumente können durch die Operatoren + (Und) / , (Oder) / * (Wildcard) verknüpft und aus einer Datei/Bibliothekselement gelesen werden (*ON %OUT=(*datei*) ...). Für ein Anwendungsbeispiel siehe Seite **535**.

Die %OUT-Bedingung ist erfüllt, wenn der Suchstring bzw. einer der Suchstrings in einer Ausgabenachricht (WROUT bzw. Ausgabeteil eines WRTRD) enthalten ist. In diesem Falle wird ein Pointer %PTR auf das Ende des gefundenen Suchstrings in der Ausgabenachricht gesetzt. Der Pointer kann dazu benutzt werden, einen Teilstring der Ausgabenachricht an eine Prozedurvariable zuzuweisen. Für weitere Informationen siehe Seite **504**.

CONTINUE Der prozedurgesteuerte Ablauf wird wieder aufgenommen. Bei einem WROUT-Aufruf wird die nächste Eingabe durch die Prozedur getätigt. Bei einem WRTRD-Aufruf wird die durch diesen Aufruf angeforderte Eingabe bereits durch die Prozedur getätigt.

CONTINUE AFTER INPUT
Der prozedurgesteuerte Ablauf wird wieder aufgenommen, nachdem der Benutzer die Eingabe zu dem aktuellen WRTRD-Aufruf getätigt hat.

SKIP *.label* Der prozedurgesteuerte Ablauf wird wieder aufgenommen. Es wird in der Prozedur ein Sprung zu der angegebenen Sprungmarke durchgeführt. Anstelle von SKIP kann auch GOTO angegeben werden.

SKIP AFTER INPUT *.label*
Der prozedurgesteuerte Ablauf wird wieder aufgenommen, nachdem der Benutzer die Eingabe zu dem aktuellen WRTRD-Aufruf getätigt hat. Es wird in der Prozedur ein Sprung zu der angegebenen Sprungmarke durchgeführt.

RETURN Es wird aus einer Connection in die Grundebene von CFS (Kommandofeld) zurückgekehrt.

RESUME Bei Eintreten der Bedingung wird die Prozedur auch in dem Falle fortgesetzt, daß im Moment keine Eingabe vom Benutzer angefordert ist. Die Möglichkeit den Lauf der Prozedur auch ohne Eingabeaufforderung fortsetzen zu können, ist vor allem bei Connections zu $CONSOLE von Bedeutung.

***ON %OUT** Die zuletzt definierte *ON %OUT-Bedingung wird wieder aktiviert. Es entfällt die erneute Interpretation der Suchbedingungen.

Eine ausführliche Beschreibung der Suchbedingung finden Sie auf Seite **309** und **313**.

***ON %TERM** CONTINUE | SKIP *.label*

Nachdem der Benutzer von der Prozedur die Kontrolle erhalten hat, prüft CFS vor jedem Eingabeschritt, ob mit der bevorstehenden Eingabe die Terminierungsabfrage

von CFS (TERM / NEW / UPDATED FILE-LIST ...) beantwortet wird. Ist dies der Fall, so ist die Bedingung *ON %TERM erfüllt und es wird der Anweisungsteil der ON-Bedingung ausgeführt.

CONTINUE Der prozedurgesteuerte Ablauf wird automatisch wieder aufgenommen vor dem Eingabeschritt, in dem die Terminierungsabfrage beantwortet wird. Die Beantwortung der Terminierungsabfrage muß durch die Prozedur vorgenommen werden.

SKIP *.label* Der prozedurgesteuerte Ablauf wird automatisch wieder aufgenommen vor dem Eingabeschritt, in dem die Terminierungsabfrage beantwortet wird. Es wird in der Prozedur ein Sprung zu der angegebenen Sprungmarke durchgeführt. Die Beantwortung der Terminierungsabfrage muß durch die Prozedur vorgenommen werden. Anstelle von SKIP kann auch GOTO angegeben werden.

***ON %TIME**=*s* RETURN

Diese Bedingung setzt ein Zeitintervall, nach dessen Ablauf CFS aus einer Connection in die Grundebene (Kommandofeld) zurückkehrt. Damit ist es selbst im Prozedurmodus möglich, aus einer "aktiven" Connection nach CFS zurückzukehren, ohne daß in der Connection eine Eingabeaufforderung besteht. So kann CFS auch bei Programmen mit freilaufenden Ausgaben bzw. bei Anwendungen ohne direkte Eingabeaufforderung (z.B. $CONSOLE) die Steuerung zurückbekommen.

s Elapsed Time in Sekunden (0 <= *s* <= 9999).

Beispiele:

Die folgende Prozedur eröffnet eine Connection zu $CONSOLE und setzt zuerst das Kommando /STA MSG und danach /STA P ab. Nach Rückkehr in die Grundebene durch **K2** + R wird die Connection durch die Prozedur automatisch abgebaut.

OC9$CONSOLE,,,R	R: Rückkehrschalter. Nach erfolgreichem Eröffnen der Konsole kehrt CFS in den Grundprozeß zurück.
xxxxx,C'sssss'	Eingabe der Benutzerkennung und des Passworts nach Rückkehr aus
CC9B	der Connection: K2 für die Konsole
*ON %TIME=1 RETURN	Nach einer Sekunde: Rückkehr in die Grundebene von CFS
STA MSG	Eingabe für Konsole
CC9B	nach Rückkehr in die Grundebene (*ON %TIME=1): erneut K2 für die Konsole generieren
*ON %TIME=1 RETURN	Timer für Rückkehr wieder initialisieren.
STA P	Eingabe an Konsole
*ON %CMD CONTINUE	Nach Rückkehr in die Grundebene von CFS durch den Benutzer soll der automatische Prozedurablauf wieder aufgenommen werden.
*DIA	Umschaltung der Eingabequelle auf Benutzerdialog
DC9	Beenden der Connection

Unter der Benutzerkennung TSOS soll eine Connection zum Hostrechner VAR2 eröffnet werden. Mit SM2 werden Statistikdaten zur Systemauslastung gesammelt.

```
/PROC N
/SYSFILE SYSDTA=(SYSCMD)
/SETSW ON=(1,4)
/EXEC $CFS
*002 FILE=NO
OC1,,VAR2/LOGON TSOS,...
EXEC $SM2
*ON %TIME=10 RETURN       nach 10 Sekunden: automatische Rückkehr in die Grund-ebene von CFS
HC SM2.STATISTIK,N        Die folgenden Ausgaben werden in die Hardcopydatei protokolliert
START                     Beginn der Statistikausgabe durch SM2. Die Statistik-informationen
                          werden im Intervall von 120 Sekunden ausgegeben
                          Nachdem nur Interesse an der ersten Ausgabe besteht, wird die
                          Connection nach 10 Sekunden verlassen.  Dies gibt die Möglichkeit,
                          neue CFS-Kommandos abzusetzen.
CC1B                      K2-Taste für Connection 1. Dadurch wird die weitere Ausgabe
                          der Statistikinformationen durch SM2 beendet.
NHC                       Ausschalten des Hardcopy-Modus.
LOGOFF                    Beenden der Connection.
END                       Beenden von CFS.
/SETSW OFF=(1,4)
/ENDP
```

*PROC-Anweisung - Allgemeine Bemerkungen

Falls eine *PROC-Anweisung in der Prozedur vorhanden ist (Syntax siehe unten), so werden dem Benutzer folgende zusätzlichen Dienste angeboten:

1) Beim Aufruf der Prozedur analysiert CFS zuerst die PROC-Anweisung der angesprochenen Prozedur und gibt eine Maske mit allen in der PROC-Anweisung aufgeführten Parametern zusammen mit ihren Standardwerten (Schlüsselwort-Parameter) aus. Diese Maske hat das gleiche Format wie diejenige Maske, die beim Starten einer BS2000 DO-Prozedur im Enter-Modus ausgegeben wird (siehe Seite 400).

   Die Ausgabe der Parameter-Maske kann unterdrückt werden, indem die Prozedur durch das Kommando: DO N,*prozedur*,... gestartet wird (N: No Prompt of Parameters).

2) Durch die Angabe *x* in der PROC-Anweisung (*PROC *x*,....) kann - ähnlich wie in BS2000 DO-Prozeduren - der Umfang der Protokollierung während des Prozedurablaufs festgelegt werden.

Falls eine *PROC-Anweisung in der Prozedur nicht vorhanden ist, so sind für den Benutzer und den Ersteller der Prozedur folgende Besonderheiten zu beachten:

1) Es findet keine Protokollierung von Prozeduranweisungen und Datensätzen statt (wie bei *PROC N).

2) Es kann keine Parameter-Maske zum Eintragen der Prozedurparameter ausgegeben werden. Alle mit einem bestimmten Wert zu versorgenden Variablen müssen entweder im DO-Kommando als Stellungs- oder Schlüsselwortparameter angegeben werden oder während des Ablaufs der Prozedur vom Benutzer durch die Anweisung *PROMPT angefordert werden.

3) Der Ersteller der Prozedur muß bezüglich der Prozedurparameter, die beim Aufruf der Prozedur den Prozedurvariablen zugewiesen werden, folgende Besonderheit beachten:

   Der erste beim Aufruf der Prozedur angegebene Stellungsparameter kann in der Prozedur nur durch den Variablen-Namen &1 angesprochen werden. Entsprechendes gilt für den zweiten Stellungsparameter &2 usw.

Syntax der *PROC-Anweisung

***PROC** *x*, (*par*, [" *text* "] ...)

Die Syntax der *PROC-Anweisung ist identisch mit der PROC-Anweisung in BS2000 DO-Prozeduren. In der *PROC-Anweisung fehlt lediglich der SUBDTA Parameter. Es sind beliebig viele Fortsetzungszeilen möglich. Als Fortsetzungszeichen ist '-' anzugeben. Im Unterschied zu BS2000 DO-Prozeduren beginnt eine Fortsetzungszeile mit '*'. In den einzelnen Zeilen einer PROC-Anweisung eingeschlossene Kommentare (".....") werden ebenfalls akzeptiert. Diese Kommentare haben eine besondere Funktion für die beim Aufruf der Prozedur ausgegebene Parameter-Maske (siehe unten).

Die PROC-Anweisung gemäß Format 1 muß als erste Anweisung in der Prozedur angegeben werden.

Erläuterung der einzelnen Parameter der PROC-Anweisung:

x — A | C | D | <u>N</u>
Steuerung des Umfangs der Protokollierung im Verlauf der Prozedur.

- A Es werden sowohl alle Prozeduranweisungen, wie auch alle an CFS zu übergebenden Datensätze am Bildschirm protokolliert.
- C Es werden nur die Prozeduranweisungen (*-Anweisungen) am Bildschirm protokolliert.
- D Es werden nur die an CFS zu übergebenden Datensätze am Bildschirm protokolliert.
- N Es werden weder Prozeduranweisungen noch Datensätze am Bildschirm protokolliert (Standard).

par — &*var* | &*var*=*wert*
Zu vereinbarender Prozedurparameter (Stellungs- oder Schlüsselwortparameter).

&*var* — Es wird ein Stellungsparameter für die Prozedur vereinbart. Ein Stellungsparameter hat keinen Wert vordefiniert. Innerhalb der Prozedur kann mit der Variablen &*var* auf den beim Aufruf der Prozedur vom Benutzer zugewiesenen Wert der Variablen Bezug genommen werden.

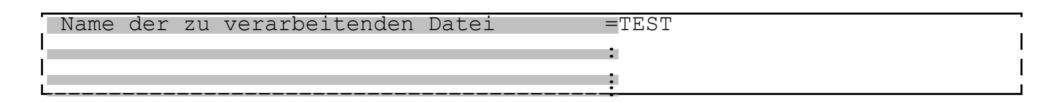
&*var*=*wert* — Es wird ein Schlüsselwortparameter für die Prozedur vereinbart. Ein Schlüsselwortparameter kann einen festen Standardwert erhalten. Beim Aufruf der Prozedur kann der Standardwert des Parameters durch die Angabe *var*=*wert* verändert werden.

wert — Standardwert des Schlüsselwortparameters. *wert* kann mit oder ohne Hochkommas angegeben werden. Enthält *wert* Sonderzeichen, wie z.B. Blanks, so muß *wert* in Hochkommas eingeschlossen werden.

"*text*" — Der hier angegebene Text wird beim DO-Aufruf in der Parameter-Maske in der Form '*var text*...' angezeigt. *var* ist dabei der Name der Variablen und *text* eine erläuternde Beschreibung zur Variablen und zu den erlaubten Eingabemöglichkeiten. Die Gesamtlänge von *var* und *text* ist auf 30 Bytes begrenzt.

Standardmäßig wird in der Bildschirmmaske für jeden Prozedurparameter eine Zeile reserviert. Dies bedeutet, daß Parameterwerte nur in der Länge von 40 Byte angegeben werden können. Durch #*n* (*n* < 4) können für einen Parameter eins bis drei zusätzliche Eingabefelder zu je 40 Byte reserviert werden. #*n* muß nach dem ersten " bzw. nach ": angegeben werden.

Beispiel: Ist in der PROC-Anweisung eine Zeile der folgenden Art enthalten:
/&DATEI=TEST, "**:#3**Name der zu verarbeitenden Datei"-
so wird dieser DO-Parameter in der Maske in folgender Form angezeigt:


```
 Name der zu verarbeitenden Datei        =TEST
                                         :
                                         :
```

"*:text*" — Der hier angegebene Text wird beim DO-Aufruf in der Parameter-Maske in der Form '*text*...' angezeigt. Der Name der Variablen wird hierbei nicht ausgegeben. *text* ist eine erläuternde Beschreibung zur Variablen und zu den erlaubten Eingabemöglichkeiten. Die Gesamtlänge von *text* ist auf 30 Bytes begrenzt.

Falls zu einem vereinbarten Parameter kein Text angegeben wurde, so erscheint der Parameter in der Parameter-Maske in der Form '*var* '.

Syntax der *PROC-Anweisung

***PROC** *x* Dieses Format der *PROC-Anweisung dient dazu, den Umfang der Protokollierung der Anweisungen am Bildschirm zu verändern. *PROC *x* kann mehrmals innerhalb der Prozedur angegeben werden und muß nicht als erste Anweisung angegeben werden.

x A | C | D | <u>N</u>

A Es werden sowohl alle Prozeduranweisungen, wie auch alle an CFS zu übergebenden Datensätze am Bildschirm protokolliert.

C Es werden nur die Prozeduranweisungen (*-Anweisungen) am Bildschirm protokolliert.

D Es werden nur die an CFS zu übergebenden Datensätze am Bildschirm protokolliert.

N Es werden weder Prozeduranweisungen noch Datensätze am Bildschirm protokolliert (Standard).

Beispiel:

Die unter dem Namen S/O in der CFS.USERLIB gespeicherte CFS-Prozedur habe folgenden Inhalt:

```
*PROC N,(&1,                     ":Nr. der zu eröffnenden Conn."  -
*        &VAR=2,                 ":Verarbeitungsrechner  <1/2>"   -
*        &JOBNAME=FRANZ,         ":Jobname für LOGON"             -
*        &USER-ID=XYZ,           ":User-Id für LOGON"             -
*        &ACCOUNT=I001,          ":Abrechnungsnummer"             -
*        &PASSW='C''...'''       ":LOGON-Passwort")
*IF &1 = ''  &1 = &NEXTCON
OC&1,,VAR&VAR/.&JOBNAME.&1  LOGON &USER-ID,&ACCOUNT,&PASSW
*WRITESYS 'Conn. &1  unter &USER-ID eröffnet  (&JOBNAME)'
```

Nach dem Aufruf DO O wird folgende Parameter-Maske ausgegeben:

```
   dd.mm.yy   hh:mm:ss          J O B - R E P O R T                TSN: ....
   COMMAND :DO CFS.USERLIB(O)

                                 D O - P A R A M S
   Nr. der zu eröffnenden Conn.  :
   Verarbeitungsrechner   <1/2>  =2
   Jobname für LOGON             =FRANZ
   User-Id für LOGON             =XYZ
   Abrechnungsnummer             =I001
   LOGON-Passwort                ='C''...'''




  You may now complete the DO-Params.                              (Cancel with K1)
```

Wert für Prozedurvariable erfragen

***PROMPT** &*var* [, '*text*'] [,L] [D]

Vom Benutzer wird die Eingabe eines Wertes für eine Prozedurvariable angefordert.

&*var* Name der Prozedurvariablen, die mit dem vom Benutzer eingegebenen Wert versorgt werden soll.

text Text für Eingabeaufforderung am Bildschirm.

Falls in der PROMPT-Anweisung kein Text angegeben ist, so wird '&VAR=' als Eingabeaufforderung ausgegeben.

L Die Eingabe für die Variable &*var* wird nicht in Großbuchstaben umgewandelt.
Standard: Die Eingabe wird in Großbuchstaben umgewandelt.

D Die Eingabe für die Variable &*var* wird in einem dunkelgesteuerten Feld angefordert.
Standard: Die Eingabe zu der Anweisung PROMPT ist sichtbar.

Latente Bedingung zurücksetzen

***RESET** [%ABEND | %CMD | %END | %IN | %OUT | %LST | %TERM | %TIME]

Diese Anweisung setzt eine zuvor definierte ON-Bedingung zurück. Die RESET-Anweisung ist notwendig, wenn eine ON-Bedingung zurückgesetzt werden soll, obwohl diese noch nicht eingetreten ist. Bei Eintritt, d.h. bei Wirksamwerden einer ON-Bedingung, wird diese automatisch zurückgesetzt. Eine ON-Bedingung einer bestimmten Kategorie (%LST, %CMD, %TERM, %END, %TIME) kann aufgrund einer Definition höchstens einmal wirksam werden.
Falls keine Bedingung angegeben ist, so werden alle ON-Bedingungen zurückgesetzt

Hinweis:

Beim Aufruf einer Prozedur werden folgende ON-Bedingungen zurückgesetzt, die durch die Startup-Datei oder eine laufende Prozedur definiert wurden: ON %CMD, %IN, %OUT, %LST, %TERM. Die in einer Startup-Datei definierten Bedingungen ON %ABEND und ON %END bleiben auch beim späteren Aufruf einer CFS-Prozedur aktiv.

Prozedurvariable über FHS-Format verändern

***RDMASK** (*format* [,*bibl*]) [TAR=*n*,] [&*var1*, &*var2*, ...]

Die Anweisung *RDMASK gibt am Bildschirm ein FHS *-Format aus, wobei die Inhalte der angegebenen Prozedurvariablen in den variablen Feldern des FHS-Formats angezeigt werden. Nach dem Senden des Formats mit der Datenübertragungstaste ENTER werden die Inhalte der modifizierbaren Maskenfelder in die angegebenen Prozedurvariablen zurückübertragen. Hierbei wird das erste Eingabefeld der Maske in die als erstes angegebene Prozedurvariable übertragen, das zweite Maskenfeld in die zweite Prozedurvariable, usw.

format Formatname der FHS-Maske in der CFSLIB.
(*format,bibl*) Formatname der FHS-Maske in der angegebenen Formatbibliothek.

TAR=*n* Tabulator rechts. Nach Ausgabe der Maske steht der Cursor standardmäßig im ersten Eingabefeld (TAR=0). Mit dem Parameter TAR=*n* (*n* = 1, 2, ...) wird der Cursor in das zweite (TAR=1), dritte (TAR=2) usw. Eingabefeld der Maske positioniert.

Der Parameter kann in der Regel weggelassen werden.

&*var1* Name der mit dem Inhalt des ersten Maskenfeldes zu füllenden Prozedurvariablen. Falls die Variable noch nicht existiert, wird sie von CFS als Leervariable neu angelegt. Es ist zu beachten, daß der Datenteil von CFS-Prozedurvariablen maximal 80 Bytes umfaßt. Hieraus ergibt sich, daß längere Maskenfelder bei der Übertragung in die Prozedurvariable gekürzt werden. Bei der Ausgabe der Maske werden die Inhalte der Prozedurvariablen in den entsprechenden Maskenfeldern dargestellt.

Falls mehr variable Felder in der Maske existieren als Prozedurvariablen angegeben wurden, so gehen die Eingaben in den überschüssigen Maskenfeldern verloren.

Hinweise:

Um die in der RDMASK-Maske betätigte Datenübertragungstaste abzufragen ist nach *RDMASK die CFS-interne Variable &RDMASKIN auszuwerten. Dazu ist die folgende Anweisung anzugeben: ***IF &RDMASKIN** = *t-bez* SKIP *.label*
Beispiel: *IF &RDMASKIN = K1 SKIP .ENDE

In der Anweisung *RDMASK dürfen als Eingabeparameter keine CFS-internen Variablen wie z.B. &DATE, &TIME angegeben werden. Ausnahme: &CFSMSG.

Vor der ersten Anwendung einer *RDMASK-Anweisung muß per File-Kommando die MFHSROUT-Bibliothek zugewiesen werden: /FILE *bibl*,LINK=MROUTLIB

Mit der Anweisung *RDMASK können nur mit FHS definierte *-Formate ausgegeben werden. Bei der Erstellung eines FHS *-Formats mit IFG sind folgende Besonderheiten zu beachten: In der Maske "Adressierungshilfen, Benutzerprofil, Standardwerte zur Einsatzvorbereitung für FHS" sind im Abschnitt "Struktur des Datenübergabebereichs" folgende Eintragungen notwendig:

getrennte Attributblöcke und Feldinhalte		: NEIN
ausgerichtet	, mit Attributfeldern	: NEIN
nicht ausgerichtet	, mit Attributfeldern	: NEIN
nicht ausgerichtet	, ohne Attributfelder	: JA

Entsprechende Angaben sind notwendig bei der Definition des Formats mit FHS-Makros.

Die Anweisung *RDMASK kann auch zum bloßen Ausgeben einer FHS-Maske in einer CFS-Prozedur verwendet werden: *RDMASK (*format,bibl*)

Die Anweisung *RDMASK kann innerhalb der Prozedur über mehrere Zeilen fortgesetzt werden. Als Fortsetzungszeichen ist '-' zu verwenden. In der Fortsetzungszeile wird ein * als erstes Zeichen erwartet.

Umschalten der Eingabequelle auf SYSCMD

***RET** SYSCMD Die *RET-Anweisung ist sinnvoll, falls CFS in einer BS2000 DO-Prozedur aufgerufen wird und in dieser per DO-Kommando mehrere CFS-Prozeduren hintereinander gestartet werden sollen.
Beispiel:

```
*002 FILE=NO
DO CFS.USERLIB(PROC1)
DO CFS.USERLIB(PROC2)
DO CFS.USERLIB(PROC3)
END
```

Damit obiger Ablauf in der gewünschten Weise funktioniert, muß in jeder der CFS-Prozeduren PROC1 bis PROC3 als letzte Anweisung *RET SYSCMD angegeben werden. Hiermit wird das Eingabemedium von SYSDTA (d.h. von der aktuellen CFS-Prozedurdatei) auf die BS2000-Prozedur umgeleitet. Von dort wird sodann das nächste CFS DO-Kommando gelesen und ausgeführt.

Wurde CFS nicht in einer DO-Prozedur aufgerufen (Procedure Level Number 0), so hat *RET SYSCMD die gleiche Wirkung wie die Anweisung *DIA (siehe unten).

Wiederaufnahme der Prozedur aus dem Dialog

RET [*datei* | *bibl* (*element*)]

Dieses Kommando, das im Feld COMMAND der Dateienliste einzugeben ist, schaltet das Eingabemedium von der Tastatur auf SYSDTA um. CFS läuft ab diesem Punkt wieder prozedurgesteuert. Das Kommando RET ist das Gegenstück zu der Prozeduranweisung *DIA.

datei | *bibl*(*element*) Durch eine Parameterangabe beim RET-Kommando wird vor dem Umschalten in den Prozedurmodus eine neue Eingabequelle für SYSDTA zugewiesen: SYSFILE SYSDTA=*datei* | SYSFILE SYSDTA=*bibl (element)*.

Falls *bibl*(*element*) angegeben wurde, so ist dadurch die auszuführende CFS-Prozedur eindeutig bestimmt. Im anderen Fall durchsucht CFS zuerst die zugewiesene USERLIB (Standard: CFS.USERLIB) nach einem Element mit dem angegebenen Namen. Ist in der USERLIB kein derartiges Element vorhanden, so wird eine Datei mit der angegebenen Bezeichnung gesucht. Zum Verändern der Standard CFS-Userlib siehe Kommando UL Seite 483.

Die automatische Wiederaufnahme einer Prozedur aufgrund bestimmter Kontextbedingungen kann erreicht werden durch die Prozeduranweisung *ON:
*ON [%END | %TERM | %ABEND | %CMD | %LST | %IN | %OUT | %INP].

Auswerten und Modifizieren von Prozedurvariablen

Die Run-Anweisung bietet die Möglichkeit, die Inhalte von CFS-Prozedurvariablen (keine internen Variablen) in vielfältiger Weise auszuwerten und zu modifizieren bzw. Formatumwandlungen mit Variablen vorzunehmen. In der Run-Anweisung werden verschiedene Programm-Module angesprochen, die eine Vielfalt einzelner Funktionen realisieren. Die Menge der zur Verfügung stehenden Run-Module wird ständig erweitert. Auch werden für bestehende Module neue Unterfunktionen angeboten.

Um den Inhalt einer CFS-internen Variablen mit der Run-Anweisung zu modifizieren, ist der Inhalt zuerst einer benutzereigenen Variablen zuzuweisen. Die benutzereigene Variable kann dann mit der Run-Anweisung modifiziert bzw. ausgewertet werden.

Aufrufformat der Run-Anweisung

***RUN** (*modul*)[*param*,][&*var1*,&*var2*, ...]

modul Name der Run-Funktion. Der Name entspricht einem Programm-Modul in der CFSLIB.

param Parameterangabe entspricht der durch den Run-Modul auszuführenden Unterfunktion.

&*var* Namen der an die Prozedur übergebenen CFS-Prozedurvariablen.
Falls eine Variable noch nicht existiert, wird sie von CFS als Leervariable angelegt. Der Run-Modul hat Zugriff auf die übergebenen Prozedurvariablen und kann auch ihren Inhalt verändern.
Der Name der Variablen kann auch indirekt in der Form &&*var* angegeben werden. In diesem Fall enthält &*var* in den Daten den Namen der eigentlich zu verarbeitenden Variablen.

Als Parameter der *RUN-Anweisung dürfen keine CFS-internen Prozedurvariablen angegeben werden (Ausnahmen: &CFSMSG und &SLABEL). Die Anweisung *RUN kann innerhalb der Prozedur über mehrere Zeilen fortgesetzt werden. Als Fortsetzungszeichen ist '-' zu verwenden. In der Fortsetzungszeile wird ein * als erstes Zeichen erwartet.

Im folgenden erhalten Sie die Beschreibung aller bisher zur Verfügung stehenden Run-Module mit ihren entsprechenden Unterfunktionen. Durch die offene Programmschnittstelle ist es darüber hinaus möglich, eigene, auf spezielle Bedürfnisse zugeschnittene Run-Module selbst zu programmieren. Die Programmschnittstelle für Run-Module ist am Ende des Abschnitts beschrieben.

***RUN (CPROMPT)** {&*var1* | '*text*'} ,&*var2*
Die Routine CPROMPT gibt den in &*var1* enthaltenen oder den direkt angegebenen String '*text*' mit TYPIO an $CONSOLE aus und erwartet vom Systembediener eine Antwort der Form tsn.text . Der Text der Antwort wird in der Variablen &*var2* zurückgeliefert. Die Routine wird solange angehalten, bis der Systembediener an der Console die PAUSE beantwortet hat.

***RUN (DATE)** ...
Die Routine DATE bietet verschiedene Operationen zur Datumsberechnung.
Als erster Parameter ist ein Operationscode anzugeben, in dem die Art der gewünschten Umrechnung/Konvertierung festlegt wird. Als zweiter Parameter ist der Name der Prozedurvariablen anzugeben, die das Eingabedatum enthält. Als optionaler Parameter kann der Name einer zweiten Prozedurvariablen angegeben werden. In dieser wird das Ergebnis der Umrechnung/Konvertierung zurückgeliefert. Falls keine zweite Prozedurvariable angegeben ist, wird das Ergebnis in der ersten Variablen geliefert.
Ein Datum kann in der Eingabevariablen in der Form *ddmmyy*[*yy*], *dd.mm.yy*[*yy*], *dd-mm-yy*[*yy*] oder *dd/mm/yy*[*yy*] angegeben werden. Falls das Ergebnis der Umrechnung wieder ein Datum ist, so wird dieses in der gleichen Schreibweise wie das Eingabedatum zurückgeliefert.

***RUN (DATE){+|-}**{*n*|&*n*},&*invar* [,&*outvar*]
Die Funktion +/- addiert/subtrahiert zu dem in &*invar* angegebenen Datum *n* Tage. *n* kann als Konstante oder als Variable angegeben werden. Das Ergebnis ist wiederum ein Datum. Falls eine Variable &*outvar* angegeben ist, wird das Ergebnis in dieser Variablen abgeliefert und &*invar* bleibt unverändert. Anderenfalls wird die Variable &*invar* entsprechend modifiziert. Fehler werden mit dem String 'ERR' angezeigt.
Beispiele:

```
*RUN (DATE)+7,&INDAT,&OUTDAT
&INDAT = '30.12.1993' --> &OUTDAT = '06.01.1994'
&INDAT = '301293'     --> &OUTDAT = '060194'
&INDAT = '30121993'   --> &OUTDAT = '06011994'
&INDAT = '32121993'   --> &OUTDAT = 'ERR'
```

***RUN (DATE)-**&*invar1*,&*invar2* [,&*outvar*]
Die Funktion - liefert als Ergebnis die Anzahl der Tage zwischen dem in der ersten (&*invar1*) und der zweiten Variablen (&*invar1*) angegebenen Datum. Beide Eingabevariablen müssen ein Datum im einem der Formate yy-mm-dd/yyyy-mm-dd/ dd.mm.yy/dd.mm.yyyy/ddmmyy/ddmmyyyy enthalten. Falls eine Variable &*outvar* angegeben ist, wird das Ergebnis in dieser Variablen abgeliefert und &*invar2* bleibt un-

verändert. Anderenfalls wird die Variable &*invar2* entsprechend modifiziert. Fehler werden mit dem String 'ERR' angezeigt.

Beispiel:
```
*RUN (DATE)-&INDAT1,&INDAT2,&OUTVAR
&INDAT1 = '31.12.93'
&INDAT2 = '01.01.93'
--> &OUTVAR = '364'
```

***RUN (DATE)DW**,&*invar* [,&*outvar*]

Die Funktion DW (Day of Week) liefert zu einem Datum die Nummer des Wochentags.

Beispiel:
```
*RUN (DATE)DW,&INDAT
&INDAT = '23.05.1994' --> '1'  (Montag)
```

***RUN (DATE)DDD**,&*invar* [,&*outvar*]

Die Funktion DDD liefert zu dem in &*invar* angegebenen Datum den laufenden Tag innerhalb des Jahres (Format ddd). Fehler werden mit dem String 'ERR' angezeigt.

Beispiel:
```
*RUN (DATE)DDD,&INDAT,&OUTDAT
&INDAT = '23.05.1994' --> &OUTDAT = '143'
```

***RUN (DATE)FREE**,&*invar* [,&*outvar*]

Die Funktion FREE liefert als Ergebnis 'Y'/'N', falls das in &*invar* angegebene Datum ein arbeitsfreier Tag/kein arbeitsfreier Tag ist. Fehler werden mit dem String 'ERR' angezeigt. Als arbeitsfreie Tage gelten Samstag, Sonntag, Feiertage sowie die in der IFREE-Datei des Terminverwaltungssystems festgelegten Tage des aktuellen Jahres. Für weitere Informationen zur IFREE-Datei siehe Seite 429.

Beispiel:
```
*RUN (DATE)FREE,&INDAT
&INDAT = '23.05.1994' --> 'Y'  (Pfingstmontag)
```

***RUN (DATE)WD**,&*invar* [,&*outvar*]

Die Funktion WD (Working Day) liefert als Ergebnis 'Y'/'N', falls das in &*invar* angegebene Datum ein Arbeitstag/kein Arbeitstag Tag ist. Fehler werden mit dem String 'ERR' angezeigt. Als Arbeitstage gelten alle Tage des Jahres außer Samstag, Sonntag, Feiertage sowie die in der IFREE-Datei des Terminverwaltungssystems festgelegten Tage des aktuellen Jahres. Für weitere Informationen zur IFREE-Datei siehe Seite 429.

Beispiele:
```
*RUN (DATE)WD,&INDAT
&INDAT = '23.05.1994' --> 'N'  (Pfingstmontag)
&INDAT = '24-05-94'   --> 'Y'
```

***RUN (DATE)WDN**,&*invar* [,&*outvar*]

Die Funktion WDN (Working Day Next) liefert zu einem Eingabedatum das Datum des nächsten Arbeitstages. Als Arbeitstage gelten alle Tage des Jahres außer Samstag, Sonntag, Feiertage sowie die in der IFREE-Datei des Terminverwaltungssystems festgelegten Tage des aktuellen Jahres. Falls das Eingabedatum bereits auf einen Arbeitstag fällt, so wird es unverändert als Ausgabedatum zurückgeliefert. Für weitere Informationen zur IFREE-Datei siehe Seite 429.

Beispiele:
```
*RUN (DATE)WDN,&INDAT,&OUTDAT
&INDAT = '23.05.1994' --> &OUTDAT = '24.05.1994'
&INDAT = '24/05/1994' --> &OUTDAT = '24/05/1994'
```

***RUN (DATE)WDP**,&*invar* [,&*outvar*]

Die Funktion WDP (Working Day Previous) liefert zu einem Eingabedatum das Datum des vorhergehenden Arbeitstages. Als Arbeitstage gelten alle Tage des Jahres außer Samstag, Sonntag, Feiertage sowie die in der IFREE-Datei des Terminverwaltungssystems festgelegten Tage des aktuellen Jahres. Falls das Eingabedatum bereits auf einen Arbeitstag fällt, so wird es unverändert als Ausgabedatum zurückgeliefert. Für weitere Informationen zur IFREE-Datei siehe Seite 429.

Beispiel:
```
*RUN (DATE)WDP,&INDAT,&OUTDAT
&INDAT = '23.05.1994' --> &OUTDAT = '20.05.1994'
```

***RUN (DATE)WDF**,&*invar* [,&*outvar*]
Die Funktion WDF (Working Day First) liefert zu einem Kalendermonat das Datum des ersten Arbeitstages in diesem Monat. Als Arbeitstage gelten alle Tage des Jahres außer Samstag, Sonntag, Feiertage sowie die in der IFREE-Datei des Terminverwaltungssystems festgelegten Tage des aktuellen Jahres. Für weitere Informationen zur IFREE-Datei siehe Seite 429.

Beispiel:
```
*RUN (DATE)WDF,&INDAT,&OUTDAT
&INDAT = '05.1994'    --> &OUTDAT = '02.05.1994'
```

***RUN (DATE)WDL**,&*invar* [,&*outvar*]
Die Funktion WDL (Working Day Last) liefert zu einem Kalendermonat das Datum des letzten Arbeitstages in diesem Monat. Als Arbeitstage gelten alle Tage des Jahres außer Samstag, Sonntag, Feiertage sowie die in der IFREE-Datei des Terminverwaltungssystems festgelegten Tage des aktuellen Jahres. Für weitere Informationen zur IFREE-Datei siehe Seite 429.

Beispiel:
```
*RUN (DATE)WDL,&INDAT,&OUTDAT
&INDAT = '05.1994'    --> &OUTDAT = '31.05.1994'
```

***RUN (DATE)WW**,&*invar* [,&*outvar*]
Die Funktion WW liefert zu dem in &*invar* angegebenen Datum die Nummer der Kalenderwoche innerhalb des Jahres (Format ww). Falls eine Variable &*outvar* angegeben ist, wird das Ergebnis in dieser Variablen abgeliefert und &*invar* bleibt unverändert. Anderenfalls wird die Variable &*invar* entsprechend modifiziert. Fehler werden mit dem String 'ERR' angezeigt.

Beispiele:
```
*RUN (DATE)WW,&INDAT
&INDAT = '23.05.1994' --> '20'
&INDAT = '01.01.1994' --> '0'
&INDAT = '03.01.1994' --> '1'
```

***RUN (DATE)WW1**,&*invar* [,&*outvar*]
Die Funktion WW1 liefert zu der in &*invar* angegebenen Kalenderwoche das Datum des ersten Tages dieser Kalenderwoche (Montag). Falls eine Variable &*outvar* angegeben ist, wird das Ergebnis in dieser Variablen abgeliefert und &*invar* bleibt unverändert. Anderenfalls wird die Variable &*invar* entsprechend modifiziert. Fehler werden mit dem String 'ERR' angezeigt.

Beispiel:
```
*RUN (DATE)WW,&INDAT
&INDAT = '201994'     --> '17.05.1994'
```

***RUN (DATE)VALID**,&*invar* [,&*outvar*]
Die Funktion VALID prüft das angegebene Datum auf Gültigkeit. Die Ausgabevariable wird auf die Werte 'Y'/'N' gesetzt, falls die Eingabevariable ein bzw. kein korrektes Datum enthält.

Beispiele:
```
*RUN (DATE)VALID,&INDAT,&RETCD
&INDAT = '30.12.1993' --> &RETCD = 'Y'
&INDAT = '301293'     --> &RETCD = 'Y'
&INDAT = '30121993'   --> &RETCD = 'Y'
&INDAT = '32121993'   --> &RETCD = 'N'
&INDAT = '29.02.94'   --> &RETCD = 'N'
```

***RUN (GETLEN)**&*var*,&*outvar*
Die Routine GETLEN liefert die Länge der Variablen &*var* in der Ausgabevariablen &*outvar* zurück.

Beispiel:
```
*RUN (GETLEN)&VAR1,&VAR1LEN
&VAR1 = 'ABCD '    --> &VAR1LEN = '5'
```

***RUN (IFUQ)**[*noopt*] [,*label-1*] ... [,*label-n*] ,&*var-1*, ... ,&*var-n* ,&*skipvar*
Mit dieser Routine ist es möglich, einen Satz von Prozedurvariablen auf Eindeutigkeit zu prüfen. Dies ist vor allem bei der Verwendung der Anweisung RDMASK von Bedeutung. In der mit RDMASK ausgegebenen Maske kann dem Anwender die Mög-

lichkeit einer Auswahl vorgegeben werden. Mit IFUQ kann auf einfache Weise ermittelt werden, wie die Prozedur nach der Auswahl durch den Benutzer fortzusetzen ist. Die Beschreibung der einzelnen Parameter der Anweisung IFUQ ist in der Inline-Dokumentation der Routine IFUQ in der Bibliothek CFS.S.LMSLIB enthalten.

***RUN (MVAR)** ...
Die Routine MVAR bietet verschiedene Optionen zur Manipulation von CFS-Variablen.

***RUN (MVAR)UPPER**,&*invar* [,&*outvar*]
Die Funktion UPPER wandelt alle Kleinbuchstaben in den Daten der Variablen &*invar* in die entsprechenden Großbuchstaben um. Falls eine Variable &*outvar* angegeben ist, wird das Ergebnis in dieser Variablen abgeliefert und &*invar* bleibt unverändert. Anderenfalls wird die Variable &*invar* entsprechend modifiziert.

Beispiel:
```
*RUN (MVAR)UPPER,&IN,&OUT
&IN = '9abcDE.,-'      --> &OUT = '9ABCDE.,-'
```

***RUN (MVAR)LOWER**,&*invar* [,&*outvar*]
Die Funktion LOWER wandelt alle Großbuchstaben in den Daten der Variablen &*invar* in die entsprechenden Kleinbuchstaben um. Falls eine Variable &*outvar* angegeben ist, wird das Ergebnis in dieser Variablen abgeliefert und &*invar* bleibt unverändert. Anderenfalls wird die Variable &*invar* entsprechend modifiziert.

Beispiel:
```
*RUN (MVAR)LOWER,&IN,&OUT
&IN = '9ABCDE.,-'      --> &OUT = '9abcde.,-'
```

***RUN (MVAR)GET-VALUE-ONLY**,&*invar* [,&*outvar*]
Die Funktion GET-VALUE-ONLY löscht in der Variablen &*invar* alle Blanks am Anfang und am Ende sowie führende Nullen am Anfang. Falls eine Variable &*outvar* angegeben ist, wird das Ergebnis in dieser Variablen abgeliefert und &*invar* bleibt unverändert. Anderenfalls wird die Variable &*invar* entsprechend modifiziert.

Beispiel:
```
*RUN (MVAR)GET-VALUE-ONLY,&IN,&OUT
&IN = '   0012   '     --> &OUT = '12'
```

***RUN (MVAR)DEL-LEADING-BLANKS**,&*var1* [,&*var2*] ...
Die Funktion DEL-LEADING-BLANKS löscht in den angegebenen Variablen alle führenden Blanks.

Beispiel:
```
*RUN (MVAR)DEL-LEADING-BLANKS,&VAR1,&VAR2
&VAR1 = '   0012   '   --> '0012   '
&VAR2 = ' ABC'         --> 'ABC'
```

***RUN (MVAR)DEL-TRAILING-BLANKS**,&*var1* [,&*var2*] ...
Die Funktion DEL-TRAILING-BLANKS löscht in den angegebenen Variablen alle Blanks am Ende.

Beispiel:
```
*RUN (MVAR)DEL-TRAILING-BLANKS,&VAR1
&VAR1 = '   0012   '   --> '   0012'
```

***RUN (MVAR)DEL-LEADING-ZEROES**,&*var1* [,&*var2*] ...
Die Funktion DEL-LEADING-ZEROES löscht in den angegebenen Variablen alle führenden Nullen.

Beispiel:
```
*RUN (MVAR)DEL-LEADING-ZEROES,&VAR1,&VAR2
&VAR1 = '0012'         --> '12'
&VAR2 = '   0012'      --> '   0012'
```

***RUN (MVAR)HEX-TO-CHAR1** [,*len*] ,&*invar* [,&*outvar*]
Die Funktion HEX-TO-CHAR1 konvertiert einen u.U. nicht abdruckbaren hexadezimalen Variableninhalt byteweise in eine abdruckbare Form. 1 <= *len* <= 4: Anzahl Bytes der Variablen &*invar*. Die Länge der Eingabevariablen sollte angegeben werden, falls diese am Ende auch binäre Nullen (X'00') enthalten kann. Falls die Variable

&*outvar* angegeben ist, wird das Ergebnis der Konvertierung dort abgelegt. Anderenfalls wird das Ergebnis in der Variablen &*invar* zurückgeliefert.

Beispiel:
```
*RUN (MVAR)HEX-TO-CHAR1,4,&VAR
&VAR = X'003FC100'    --> '003FC100'
```

***RUN (MVAR)HEX-TO-CHAR2**,&*invar* [,&*outvar*]

Die Funktion HEX-TO-CHAR2 interpretiert den abdruckbaren Inhalt der Variablen &*invar* als hexadezimale Zahl und wandelt diese in die Dezimaldarstellung um. Falls eine Variable &*outvar* angegeben ist, wird das Ergebnis in dieser Variablen abgeliefert und &*invar* bleibt unverändert. Anderenfalls wird die Variable &*invar* entsprechend modifiziert.

Beispiel:
```
*RUN (MVAR)HEX-TO-CHAR2,&VAR
&VAR = '003FC100'     --> '4178176'
```

***RUN (MVAR)HEX-TO-CHAR3** [,*len*] ,&*invar* [,&*outvar*]

Die Funktion HEX-TO-CHAR3 ist eine Kombination der Funktionen HEX-TO-CHAR1 und HEX-TO-CHAR2

Beispiel:
```
*RUN (MVAR)HEX-TO-CHAR3,4,&VAR
&VAR = X'003FC100'    --> '4178176'
```

***RUN (MVAR)CHAR-TO-HEX1** [,*len*] ,&*invar* [,&*outvar*]

Die Funktion CHAR-TO-HEX1 konvertiert einen abdruckbaren Variableninhalt byteweise in den entsprechenden hexadezimalen Wert.
1 <= *len* <= 4: Länge des hexadezimalen Ausgabestrings. Die Länge des Ausgabewertes sollte angegeben werden, falls führende Nullen (X'00') eingefügt werden sollen. Falls die Variable &*outvar* angegeben ist, wird das Ergebnis der Konvertierung dort abgelegt. Anderenfalls wird das Ergebnis in der Variablen &*invar* zurückgeliefert.

Beispiel:
```
*RUN (MVAR)CHAR-TO-HEX1,3,&VAR
&VAR = '12C1'         --> X'0012C1'
```

***RUN (MVAR)CHAR-TO-HEX2**,&*invar* [,&*outvar*]

Die Funktion CHAR-TO-HEX2 wandelt die in der Variablen &*invar* enthaltene Dezimalzahl in die entsprechende hexadezimale Zahl um. Falls eine Variable &*outvar* angegeben ist, wird das Ergebnis in dieser Variablen abgeliefert und &*invar* bleibt unverändert. Anderenfalls wird die Variable &*invar* entsprechend modifiziert.

Beispiel:
```
*RUN (MVAR)CHAR-TO-HEX2,&VAR1,&VAR2
&VAR1 = '4801'        --> &VAR2 = '12C1'
```

***RUN (MVAR)CHAR-TO-HEX3** [,*len*] ,&*invar* [,&*outvar*]

Die Funktion CHAR-TO-HEX3 ist eine Kombination der Funktionen CHAR-TO-HEX1 und CHAR-TO-HEX2

Beispiel:
```
*RUN (MVAR)CHAR-TO-HEX3,4,&VAR
&VAR = X'4801'        --> X'000012C1'
```

***RUN (MVAR)FILL-STRING**,*len*, &*invar* [,&*outvar*]

Die Funktion FILL-STRING füllt den Inhalt einer Variablen rechts bis zu der gewünschten Länge *len* mit Blanks (' ') auf. 2 <= *len* <= 80. Falls die Variable &*outvar* angegeben ist, wird das Ergebnis dort abgelegt. Anderenfalls wird das Ergebnis in der Variablen &*invar* zurückgeliefert.

Beispiel:
```
*RUN (MVAR)FILL-STRING,10,&VAR
&VAR = '   0012'      --> '   0012   '
```

***RUN (MVAR)FILL-STRING-RIGHT**,*len* [,*fill*] ,&*invar* [,&*outvar*]

Die Funktion FILL-STRING-RIGHT richtet den Inhalt einer Variablen in der Länge *len* rechts aus, indem links vorgegebene Füllzeichen *fill* eingefügt werden. 2 <= *len* <= 80. Falls die Variable &*outvar* angegeben ist, wird das Ergebnis dort abgelegt. Anderenfalls wird das Ergebnis in der Variablen &*invar* zurückgeliefert.

Für *fill* kann ein Zeichen im Format [C]'*x*' angegeben werden. Standard: C'␣'

Beispiel:
```
*RUN (MVAR)FILL-STRING,6,&VAR1,&VAR2
&VAR1 = '0012'        --> &VAR2 = '  0012'
```

***RUN (MVAR)ADD-ARRAY**, &*invar1*, &*invar2*, &*outvar*

Die Funktion ADD-ARRAY führt eine spaltenweise Addition aller Felder von zwei gleich strukturierten Variablen aus. Die zu addierenden Felder können auch Zahlen der Form 999.99 oder 999,99 enthalten. Die Verarbeitung der Felder erfolgt von rechts nach links. Deshalb müssen alle Spalten in den zwei Eingabevariablen rechtsbündig gleich ausgerichtet sein. In der Variablen &*outvar* wird das Ergebnis abgelegt.

Beispiel:
```
*RUN (MVAR)ADD-ARRAY,&VAR1,&VAR2,&SUMME
&VAR1 = '  12     1,35    99'
&VAR2 = '   2    12,65    01'
&SUMME= '  14    14,00   100'
```

***RUN (MVAR)SUB-ARRAY**, &*invar1*, &*invar2*, &*outvar*

Die Funktion SUB-ARRAY führt eine spaltenweise Subtraktion aller Felder von zwei gleich strukturierten Variablen aus (&*invar1* - &*invar2*). Die zu subtrahierenden Felder können auch Zahlen der Form 999.99 oder 999,99 enthalten. Die Verarbeitung der Felder erfolgt von rechts nach links. Deshalb müssen alle Spalten in den zwei Eingabevariablen rechtsbündig gleich ausgerichtet sein. In der Variablen &*outvar* wird das Ergebnis abgelegt.

Beispiel:
```
*RUN (MVAR)SUB-ARRAY,&VAR1,&VAR2,&DIFF
&VAR1 = '  12    12,35    99'
&VAR2 = '   2     1,65    01'
&DIFF = '  10    10,70    98'
```

***RUN (MVAR)TIME-DIFF**, &*invar1*, &*invar2*, &*outvar*

Die Funktion TIME-DIFF subtrahiert von der in &*invar2* angegebenen Zeit (HH:MM:SS) die in &*invar1* enthaltene Zeit. Das Ergebnis wird als HH:MM:SS in die Variable &*outvar* geschrieben.

Beispiel:
```
*RUN (MVAR)TIME-DIFF,&VAR1,&VAR2,&DIFF
&VAR1 = '11:54:22'
&VAR2 = '12:25:21'
&DIFF = '00:30:59'
```

***RUN (MVAR)TIME-DIFF-TEXT**, &*invar1*, &*invar2*, &*outvar*

Die Funktion TIME-DIFF-TEXT subtrahiert von der in &*invar2* angegebenen Zeit (HH:MM:SS) die in &*invar1* enthaltene Zeit. Das Ergebnis wird in textlicher Aufbereitung in die Variable &*outvar* geschrieben.

Beispiel:
```
*RUN (MVAR)TIME-DIFF-TEXT,&VAR1,&VAR2,&DIFF
&VAR1 = '11:54:22'
&VAR2 = '12:25:21'
&DIFF = '30 minutes, 59 seconds'
```

***RUN (SETLEN)**&*var1* [,&*var2*] ...

Die Routine SETLEN kürzt Prozedurvariablen &*var1*, &*var2* usw. in der Weise, daß von Rechts her alle Blanks durch X'00' ersetzt werden.

Anwendungsfall: Bei RDMASK werden die über die FHS-Maske eingelesenen Prozedurvariablen in der Länge des jeweiligen Maskenfeldes mit Blanks aufgefüllt. Die so eingelesenen Variablen können durch SETLEN gekürzt werden.

***RUN (RDJOBPAR)** [*col* [,*len*],] &*var*

Die Routine RDJOBPAR liefert in der Variablen &*var* die im LOGON- bzw. ENTER-Kommando vereinbarten Job-Parameter (/LOGON ...,JOB-PAR='...'). Als Job-Parameter kann einer Task ein String in der Länge 1 bis 128 mitgegeben werden. Mit *RUN (RDJOBPAR) kann dieser String zur Steuerung von CFS-Prozeduren in Dialog- und Enter-Tasks verwendet werden. Durch die optionalen Angaben *col* und *len* wird ab einer bestimmten Spalte (1 <= *col* < 128) und in einer vorgegebenen Länge (1 <= *len* <= 80) gelesen.

RUN (SVAR)such*,&*var*,&*rcvar* [,&*findvar1* [,&*findvar2*]]

Die Routine SVAR prüft, ob der Inhalt der Prozedurvariablen &*var* eine in CFS-Syntax angegebene allgemeine Suchbedingung *such* (siehe Seite **309**/**313** erfüllt. Die Suchbedingung kann direkt oder über eine Prozedurvariable angegeben werden.

In der als zweites angegebenen Prozedurvariablen &*rcvar* wird ein Returncode hinterlegt, der über das Ergebnis der Suche informiert.

'Y'|'N'	Das Suchitem wurde gefunden/nicht gefunden.
''	Syntaxfehler im Suchargument.

In die optionale Variable &*findvar1* wird im Falle von &*rcvar*='Y' der auf den gefundenen Suchbegriff folgende Teil der Variablen &*var* eingetragen.

In die optionale Variable &*findvar2* wird im Falle von &*rcvar*='Y' der vor dem gefundenen Suchbegriff stehende Teil der Variablen &*var* eingetragen. &*findvar2* kann nur zusammen mit &*findvar1* angegeben werden.

Das Quellprogramm der Routine SVAR ist in der PLAM-Bibliothek CFS.S.LMSLIB enthalten und kann vom Benutzer auf eigene Bedürfnisse angepaßt werden.

***RUN (URLOUT)**HTTP [,*type*]

Die Routine URLOUT arbeitet mit unserem Produkt **URLSERVER** zusammen.

An den mit dem URLSERVER verbundenen Internet-Browser wird ein HTTP-Header mit dem angegebenen Content-Type gesandt.

Der gesendete HTTP-Header beinhaltet folgende Daten:

```
HTTP/1.0 200 OK
Content-Type: type
```

(leerzeile)

type — Der Content-Type kann als CFS-Variable &var oder als String angegeben werden.

Fehlt der Parameter *type*, wird der Content-Type "text/html" verwendet.

Beispiele:

```
*&TYPE='text/html'
*RUN (URLOUT)HTTP,&TYPE
*RUN (URLOUT)HTTP,text/plain
*RUN (URLOUT)HTTP,'text/plain'
```

***RUN (URLOUT)**HREF ,*url*

An den Internet-Browser wird ein HTTP-Header mit dem Status-Code 302 (andere URL-Adresse anfordern) gesendet.

Diese Anweisung ist nur sinnvoll, wenn sie als einzige URLOUT-Anweisung am Ende einer CFS-Prozedur angegeben wird. D.h. es wird zuerst eine Datei erzeugt und dann die Anzeige dieser Datei mit *RUN (URLOUT)HREF gestartet. Dabei können alle Funktionen des URL-Servers benutzt werden, wie z.B. VIEW, XVIEW, EDTW, PDF.

url — Die URL-Adresse kann als CFS-Variable &var oder als String angegeben werden.

Beispiele:

```
*&GOTO='../PDF/TEST.LST'
*RUN (URLOUT)HREF,&GOTO
*RUN (URLOUT)HREF,../PDF/TEST.LST
```

***RUN (URLOUT)**DATA [*,string*] [,&*var1*, &*var2*, ...]

An den Internet-Browser wird der angegebene konstante String und/oder der Inhalt der CFS-Variablen gesendet. Die Daten werden von EBCDIC nach ISO8859 konvertiert.

string Text mit Daten und HTML-Tags, die an den Internet-Browser gesendet wird. Der String kann in Hochkommas eingeschlossen werden.

&var1 ,&var2 ...

Die angegebenen Variablen können Text und HTML-Tags enthalten und werden hintereinander und direkt im Anschluss an einen evtl. angegebenen String an den Internet-Browser gesendet. Eine Variable kann bis zu 80 Bytes Text enthalten. Insgesamt darf der Inhalt aller Variablen die Länge von 4096 Bytes nicht überschreiten.

Beispiele:
```
*&VAR1='Daten Zeile 1<BR>'
*&VAR2='Daten Zeile 2<BR>'
*RUN (URLOUT)DATA,'<BR>',&VAR1,&VAR2
*RUN (URLOUT)DATA,'FSTAT für Datei: '
*RUN (URLOUT)DATA,&FILE
*RUN (URLOUT)DATA,<BR>
```

***RUN (URLOUT)**DATAFILE *,file*

An den Internet-Browser wird der Inhalt der angegebenen Datei gesendet. Die Daten werden von EBCDIC nach ISO8859 konvertiert.

Die Datei muß Daten im HTML-Format enthalten oder vor der Anweisung *RUN (URLOUT)DATAFILE muß der HTML-Tag <PRE> ausgegeben werden oder es muß als Content-Type "text/plain" verwendet werden.

file Name der Datei, deren Inhalt ausgegeben werden soll. Der Name der Datei kann auch über eine CFS-Variable angegeben werden.

Beispiel:
```
*RUN (URLOUT)DATAFILE,TESTHTML
```

***RUN (SYSOUT)** [EXTEND] ,&*sysout,* &*oldsysout*

Diese Funktion bietet die Möglichkeit, in einer Prozedur SYSFILE SYSOUT auf eine neue Datei zuzuweisen und den Namen der bisherigen SYSOUT-Datei in einer CFS-Variablen zu speichern. Damit kann nach der Erstellung von SYSOUT-Daten wieder die bisherige SYSOUT-Datei zugewiesen werden.

EXTEND Die neu zugewiesene Sysout-Datei wird mit OPEN-MODE=*EXTEND geöffnet.

&*sysout* CFS-Variable, die den Namen der neuen Sysout-Datei enthält.

&*oldsysout* CFS-Variable, in die der Name der alten Sysout-Datei übertragen werden soll.

Beispiel:
```
*&SYSOUT='#OUT'
*RUN (SYSOUT),&SYSOUT,&OLDSYSOUT
*/SHOW-FILE-ATTR &FILE,INF=*A
*RUN (SYSOUT)EXTEND,&OLDSYSOUT,&SYSOUT
*RUN (URLOUT)DATAFILE,&SYSOUT
```

***RUN (WRITECON)**&*var*

Die Routine WRITECON gibt den Inhalt von &*var* an der Console aus (TYPE).

Allgemeines Aufrufformat für Run-Module

***RUN** (*modul* [, *bibl*]) [*param,*] [&*var1*, &*var2*, ...]

Die Anweisung *RUN ruft ein benutzereigenes Unterprogramm auf.

modul Name des Benutzermoduls in der CFSLIB.

(*modul,bibl*) Der Benutzermodul ist in der angegebenen Bibliothek gespeichert.

param Parameterangabe zur Auswertung durch den Benutzermodul.

&var1 Name der ersten, an die Prozedur übergebenen CFS-Prozedurvariablen.
Falls die Variable noch nicht existiert, wird sie von CFS als Leervariable neu angelegt. Das Benutzermodul hat Zugriff auf die Prozedurvariable und kann auch ihren Inhalt verändern.

Programmschnittstelle für benutzereigene Run-Module

Beim Aufruf eines Run-Moduls zeigt das Mehrzweckregister R1 auf eine Parameterleiste mit folgender Struktur:

```
DC  A(param)     Steuerungsparameter (siehe oben) bzw. Blank, falls param
                 beim Aufruf des Benutzermoduls nicht angegeben wurde.
DC  A(varlist)   varlist: DC  A(var1)
                          DC  A(var2)
                              .....
                          DC  A(var-x)
                          DC  A(0)      Endekennzeichen
```

Eine CFS-Prozedurvariable *var-x* ist eine Datenstruktur mit folgendem Aufbau:

```
DC  CL20'var-name'       Name der Prozedurvariablen
DC  XL80'00'             Wert der Prozedurvariablen (anfangs X'00')
```

Als Wert der Prozedurvariablen wird die Zeichenfolge vom Beginn bis zum ersten Zeichen X'00' angesehen. Eine noch nicht initialisierte Prozedurvariable (XL80'00') besitzt somit als Wert den Leerstring ''.

Über Prozedurvariable ist eine Kommunikation in der Art CFS-Prozedur --> Benutzermodul --> CFS-Prozedur möglich.

Der Inhalt des Mehrzweckregisters R15 wird nach dem Rücksprung aus dem Benutzermodul nach CFS nicht ausgewertet.

Nächste Maske/WRTRD-Nachricht am Bildschirm ausgeben

***SCRUPD** Screen-Update. Während des Ablaufs von CFS-Prozeduren wird die Anzeige von WRTRD-Nachrichten zur Anforderung der nächsten Eingabe unterdrückt. Es werden z.B. die CFS-Masken, in denen der Benutzer im Dialog seine Eingaben tätigt, nicht ausgegeben. Mit der Anweisung *SCRUPD wird diese Unterdrückung der Maskenausgabe für die nächste WRTRD-Nachricht ausgeschaltet. *SCRUPD bewirkt, daß die Nachricht (Maske) zur Anforderung der nächsten Eingabe mit WROUT am Bildschirm ausgegeben wird. *SCRUPD vor einer *002-Anweisung ist unzulässig.

Die Anweisung *SCRUPD kann z.B. genutzt werden, wenn in CFS-Prozeduren Kommandos vor ihrer Ausführung vom Benutzer abgeändert werden können (Anweisung *STEP, siehe unten). Dem Benutzer wird durch Anzeige der Bildschirmmaske mitgeteilt, auf welche Datenobjekte das vorgeschlagene Kommando angewendet wird.

Senden einer Nachricht an $CONSOLE

***SEND** '*text*' Der in Hochkommas eingeschlossene Text wird an eine unter CFS eröffnete Connection zu $CONSOLE am eigenen Host gesendet. Der Text kann auch CFS-Prozedurvariable enthalten. Beispiel: *SEND '&TSN..N'
Für ein Anwendungsbeispiel für die *SEND-Anweisung siehe Seite 535.

SENDn* '*text*' Der in Hochkommas eingeschlossene Text wird an die unter CFS eröffnete Connection mit der Nummer *n* gesendet. Als Applikationen, an die der Text gesendet wird, sind $CONSOLE auf beliebigen Host-Rechnern sowie $VMCONS erlaubt.

Prozedurspezifische Steuerzeichen ändern

***SET** [TAB=*x*] [, NOC=*y*] [, PAR=*p*] [, EM=Y|N] [, CASE=Y|N|X] [, PAR00D=Y|N]

Ändern von Steuerzeichen, die in der Prozedursprache von CFS eine besondere Funktion besitzen.

TAB Tabulatorzeichen (Standardwert: C'~')

NOC Beliebigkeitszeichen (Standardwert: C'|').

PAR Parameterzeichen für Variable in CFS-Prozeduren (Standardwert: C'&').
Weitere Informationen zu CFS-Prozedurvariablen finden Sie auf Seite **491**.

EM Behandlung von CFS-Kommandos.

EM=Y Im Prozedurmodus werden Kommandoeingaben mit der Endemarke abgeschlossen.

EM=N Im Prozedurmodus werden Kommandoeingaben wie im Dialog mit ENTER, d.h. ohne Endemarke gesendet. (Standard)

CASE Behandlung von Kleinbuchstaben bei Vergleichen mit der Anweisung *IF &*var*=...

CASE=Y Kleinbuchstaben werden beim Vergleich zweier Variablen bzw. einer Variablen mit einem konstanten Wert berücksichtigt.

CASE=N Kleinbuchstaben werden beim Vergleich zweier Variablen bzw. einer Variablen mit einem konstanten Wert nicht berücksichtigt, d.h. sie werden vor dem Vergleich in die entsprechenden Großbuchstaben umgewandelt. (Standard)
Im Falle von CASE=N ist bein Vergleich von Variablen folgendes zu beachten. Beispiel: `&VAR1='1234 '`
`*IF &VAR1='1234'`
führt zu einem negativen Vergleichsergebnis.
Bei CASE=X (s.u.) würde dieselbe IF-Anweisung zu einem positiven Vergleichsergebnis führen.

CASE=X wie CASE=N, jedoch werden auch die Zeichen X'00' in X'40' (Blank) umgewandelt. Damit kann beim Vergleich von Variablen der bis Ende 1996 gültige kompatible CFS-Modus hergestellt werden. Es ist in diesem Modus jedoch nicht möglich eine leere Variable von einer Variablen mit Blanks als Inhalt zu unterscheiden.
Beispiel: `&VAR1=' ',  &VAR2=''`
`*IF &VAR1 = '_' ....` --> Ergebnis positiv
`*IF &VAR2 = '_' ....` --> Ergebnis positiv (nicht korrekt)
`*IF &VAR1 = &VAR2 ..` --> Ergebnis positiv (nicht korrekt)
Der kompatible Modus CASE=X kann auch über den Modul CFSMAIN für das Rechenzentrum voreingestellt werden.

PAR00D Behandlung von Nachrichten mit Auto-DUE Steuerzeichen in Prozeduren.

PAR00D=N Diese Einstellung ist nur für seltene Sonderfälle geeignet.
Beispiel: Automatische Steuerung von Programmen wie TELNET in CFS-Connections. Ein Anwendungsbeispiel finden Sie im Element J/TELNET in der ausgelieferten PLAM-Bibliothek CFS.USERLIB.BEISPIEL.

PAR00D=Y Standardwert. Dieser muß normalerweise nicht verändert werden.

Steuerzeichen in CFS-Prozeduren

Das Zeichen C'~' gilt per Voreinstellung als Tabulatorzeichen für die in einer Prozedur simulierte Eingabe einer CFS-Maske (Dateienliste/Display-Maske). Das Tabulatorzeichen bewirkt, daß die folgenden Zeichen im ersten Action-Feld (Eingabespalte 72) bzw. im nächsten Action-Feld/im Modify-Bereich weitergeführt werden.

Das Zeichen C'|' gilt als Beliebigkeitszeichen (Not Overwrite Character) für die in einer CFS-Prozedur simulierte Eingabe einer CFS-Maske (Dateienliste/Display-Maske). Das Beliebigkeitszeichen bewirkt, daß die entsprechende Stelle aus der im Prozedurmodus simulierten Bildschirmmaske für die Eingabe übernommen wird.

Bei der Erstellung von CFS-Prozeduren empfiehlt es sich, zu Beginn den Tabulator und das Beliebigkeitszeichen auf vom Benutzer selbst vorgegebene Zeichen zu legen. Falls die Zeichen '%' und '#' in den Daten der Prozedur nicht enthalten sind, wird mit der Anweisung *SET TAB=%,NOC=# das Zeichen für den Tabulator auf '%' und das Beliebigkeitszeichen auf '#' gesetzt.

Beispiel:

```
%%A##BBB
```

Annahme: CFS befindet sich im Modify-Modus. Die prozedurspezifischen Steuerzeichen wurden mit *SET TAB=%,NOC=# auf % bzw. # gesetzt.
Aufgrund der Prozedureingabe wird die erste Stelle des Modify-Bereichs mit dem Zeichen 'A' überschrieben ('%%' überspringt das Kommando und das Action-Code Feld der CFS-Maske). Die Stellen 2 und 3 des Modify-Bereichs werden nicht verändert (Übernahme der in der Maske stehenden Zeichen). Die Stellen 4 bis 6 werden mit dem Zeichen 'B' überschrieben.

Prozeßschalter verändern

***SETSW** { ON | OFF } = (*n1*, ...)

Setzen oder Rücksetzen von Prozeßschaltern. Syntax analog /SETSW-Kommando des BS2000. In einer Anweisung ist nur SETSW ON oder SETSW OFF für ein bzw. mehrere Schalter möglich.

Verzweigung innerhalb der Prozedur

***SKIP** .*label*

Unbedingter Sprung zu der angegebenen Sprungmarke. Sprünge in Richtung Prozeduranfang (Rückwärtssprünge) sind ebenfalls möglich. Wird die Sprungmarke nicht gefunden, so wird der Ablauf der Prozedur beendet. Anstelle von .*label* kann auch eine Prozedurvariable angegeben werden, die das Sprungziel als Text enthält (*SKIP &*var*). Für ein Anwendungsbeispiel hierzu siehe Seite **535**.
Anstelle von SKIP kann auch GOTO angegeben werden.

Eingabevorschlag für Benutzer

***STEP** *text*

Der angegebene Text wird am Bildschirm ausgegeben und kann vom Benutzer verändert werden. Der so veränderte oder auch unverändert abgesendete Text wird als Eingabe an CFS weitergereicht. Wird der Eingabevorschlag mit der K1-Taste beantwortet, so wird das Kommando nicht ausgeführt und die weitere Abarbeitung der Prozedur beendet.

Anhalten der Prozedur bis Quittierung durch Benutzer

***STOP** [*'text'*] Der in Hochkommas eingeschlossene Text wird am Bildschirm ausgegeben. Der Benutzer muß die Nachricht mit ENTER quittieren, bevor die Prozedur weiterläuft. Der Text kann bis zu 2040 Bytes lang sein und kann auch logische Steuerzeichen enthalten (Ausgabe im LINE-Mode). Falls kein Text angegeben wurde, so wird der Standardtext "PLEASE ACKNOWLEDGE" ausgegeben.

Inhalt von Prozedurvariablen anzeigen/in Datei sichern

***SV** [&*var*] [, *datei*] [, E] [, NE]

&*var* Der Inhalt der Prozedurvariablen soll am Bildschirm angezeigt bzw. in eine Datei gesichert werden. Falls &*var* nicht angegeben ist, werden die Namen und Inhalte aller Prozedurvariablen ausgegeben bzw. in eine Datei geschrieben (s.u.).

datei Eine bestimmte bzw. alle in der Prozedur angelegten Variablen (*&*var*= ...) werden mit ihren aktuellen Werten in die angegebene Datei geschrieben und können damit in anderen Prozeduren wieder geladen werden (*LV-Anweisung).
Falls *datei* nicht angegeben ist, werden die Namen und Inhalte der Variablen am Bildschirm angezeigt. Das Komma vor dem Dateinamen kann weggelassen werden, falls &*var* nicht angegeben wurde.

E Die Datei wird im Extend-Modus eröffnet.

NE Not Empty. Es werden nur Variablen in die Datei geschrieben, die einen echten Inhalt besitzen und somit keine Leervariablen sind.

Die in der SV-Datei gespeicherten Sätze haben folgendes Format:
20 Byte Variablenname; 1 bis 80 Byte Variableninhalt. Der Variableninhalt kann (muß aber nicht) mit Nil-Zeichen (X'00') auf 80 Byte aufgefüllt sein.

Prozedurablauf für bestimmte Zeit anhalten

***WAIT** [*n*] [, *.label*]

n Der Prozedurablauf wird für *n* Sekunden angehalten. Falls *n* nicht angegeben: unbegrenzte Wartezeit.

.label Nach Ablauf der Wartezeit erfolgt eine Verzweigung der Prozedur zum angegebenen Label (*SKIP *.label*).

Die in der Anweisung *WAIT angegebene Wartezeit kann vom Benutzer durch Betätigen der K2-Taste vorzeitig beendet werden. In diesem Fall wird keine Verzweigung zu einem evtl. angegebenen Sprungziel (*.label*) durchgeführt.

Ausgabe einer Nachricht an den Bildschirm

***WRITE** [*'text'*|CLEAR] [, *datei* [,O|E|N]]

Der in Hochkommas eingeschlossene Text wird am Bildschirm bzw. in eine Datei ausgegeben.

text Der Text kann feste Strings und CFS-Prozedurvariablen oder logische Steuerzeichen enthalten (WROUT-Ausgabe im LINE-Mode). Falls *text* nicht angegeben wurde, so gibt CFS eine Leerzeile aus.
Falls im Text der Inhalt von CFS-Prozedurvariablen ausgegeben werden soll, so ist zu beachten, daß Konstrukte von der Form &*var*(*x*,*y*), d.h. Variablen mit Offset und Längenangabe nicht erlaubt sind. Ggf. ist dazu eine Hilfsvariable zu definieren: *&*hilf*=&*var*(*x*,*y*). Diese Hilfsvariable kann im Write-Text verwendet werden.

CLEAR Es wird eine Ausgabe getätigt, die den Bildschirm löscht.

datei Name der Datei (Fcbtype=SAM), in die der Text geschrieben wird.

O Die Datei wird im Output-Modus eröffnet. Eine bestehende Datei gleichen Namens wird überschrieben. Die Datei wird nach dem Schreiben des Satzes geschlossen.

E Die Datei wird im Extend-Modus eröffnet. Der zu schreibende Text wird am Ende der Datei angefügt. Die Datei wird nach dem Schreiben des Satzes geschlossen.

N Die Datei wird im Extend-Modus eröffnet und bleibt nach dem Schreiben des Satzes weiterhin geöffnet. Mit diesem Schalter wird verhindert, daß die Datei bei jedem Schreiben eines Satzes um eine Pam-Seite anwächst.

Ausgabe einer Nachricht in die Systemzeile

***WRITESYS** '*text*' Der in Hochkommas eingeschlossene Text wird in der Systemzeile des Bildschirms ausgegeben.

Durch die Anweisung *WRITESYS '' kann der Ausgangszustand der Systemzeile ("LTG") wieder hergestellt werden.

Starten einer Prozedur

DO [N,] [*datei* | *element* | *bibl*(*element*) | %*name* | %?] [, *param*, ...]

Die angesprochene Prozedur enthält eine optionale PROC-Anweisung mit darin vereinbarten Prozedurparametern. Normalerweise wird beim Aufruf einer derartigen Prozedur zuerst die Parameter-Maske ausgegeben, in der die Prozedurparameter eingegeben werden können.

N Durch die Eingabe DO N,... wird die Ausgabe der Parameter-Maske unterbunden. Es werden die Prozedur-Parameter verwendet, so wie sie im DO-Kommando angegeben wurden (siehe unten), bzw. wie sie in der PROC-Anweisung vordefiniert sind.

datei Die CFS-Prozedur ist in der angegebenen Datei gespeichert.

element Die Prozedur ist in dem angegebenen Element der CFS.USERLIB bzw. in der zentralen USERLIB (siehe Hinweise) enthalten. In PLAM-Bibliotheken gespeicherte Prozeduren müssen den Elementtyp S (Source) besitzen.

bibl(*element*) Die Prozedur ist in dem Element der angegebenen Bibliothek enthalten.

%*name* Es wird auf einen Parameter-Set, d.h. auf den Namen einer Prozedur und auf eine bestimmte Parameterbelegung Bezug genommen, der zu einem früheren Zeitpunkt mit dem Kommando SP %*name* unter dem angegebenen Kürzel %*name* gesichert wurde. Eine ausführliche Beschreibung des SP-Kommandos und der Verwendung von %name finden Sie auf Seite 403.

%? Es werden alle in Frage kommenden Parametersets in einer Maske angezeigt. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

param Wahlweise Angabe von Werten für CFS-Prozedurvariable. Wie bei DO-Prozeduren im BS2000 können sowohl Stellungs-, als auch Schlüsselwortparameter angegeben werden. Von welcher Art ein Parameter ist, wird in der *PROC-Anweisung der Prozedur festgelegt. Parameter können nicht zusammen mit %*name* angegeben werden.

Stellungsparameter: Es ist lediglich der Wert des gewünschten Parameters anzugeben. Bei Stellungsparametern ist auf deren richtige Reihenfolge zu achten. Soll z.B. nur der zweite Stellungsparameter mit einem Wert versorgt werden, so ist ein zusätzliches Komma bei der Parameterangabe einzufügen: DO *prozedur,,wert*.

Schlüsselwortparameter: Es ist der Name des Parameters, gefolgt von einem = Zeichen und dem gewünschten Parameterwert anzugeben. Der Name des Parameters und sein Standardwert können in der *PROC-Anweisung der Prozedur festgelegt werden.

Hinweise:

Das Kommando DO wird von CFS in folgender Weise abgearbeitet:
Falls *bibl*(*element*) bzw. %*name* angegeben wurde, so ist dadurch die auszuführende CFS-Prozedur eindeutig bestimmt. In allen anderen Fällen wird gemäß der folgenden Hierarchie nach der Prozedur gesucht:

a) CFS sucht in der zugewiesenen Userlib (Standard: CFS.USERLIB) nach einem Element mit dem angegebenen Namen.

b) Es wird in der zentralen USERLIB nach einem Element mit dem angegebenen Namen gesucht. Die zentrale USERLIB kann mit /FILE ...,LINK=USERLIB zugewiesen werden oder wird als Bibliothek mit dem Namen $*user-id*.CFS.USERLIB. STD gesucht. $*user-id* = Benutzerkennung in der die Programmphase CFS gespeichert ist.

c) Es wird eine Datei mit der angegebenen Bezeichnung gesucht.

Wenn a), b) und c) nicht zu einem Erfolg geführt haben, wird eine Fehlermeldung ausgegeben. Zum Verändern der Standard CFS-Userlib siehe Kommando UL, Seite 483.

Wird das DO-Kommando ohne irgendwelche Zusätze angegeben, so wird die zuletzt verwendete Prozedur wieder aufgerufen.

Enthält ein Parameterwert Leerzeichen (Blanks) oder bestimmte Sonderzeichen, wie z.B. Hochkommas, so ist der Parameterwert in Hochkommas einzuschließen.
Enthält ein Parameterwert Hochkommas, so müssen diese verdoppelt werden.

Falls die angesprochene Prozedur keine *PROC-Anweisung enthält, so ist kein maskengesteuertes Prompting der Prozedurparameter möglich. Die Prozedur kann nicht über %*name* angesprochen werden.

Falls beim Aufruf einer Prozedur ohne *PROC-Anweisung Stellungsparameter angegeben wurden, so können diese innerhalb der Prozedur nur über die Namen &1 (erster Stellungsparameter), &2 (zweiter Stellungsparameter), usw. angesprochen werden.

Testen von Prozeduren

Falls eine Prozedur nicht in der Art und Weise abläuft, wie der Benutzer es erwartet, so sind zwei verschiedene Arten der Ablaufverfolgung möglich.

a) Einschalten der Protokollierung aller in der Prozedur ausgeführten Anweisungen durch Einfügen der Anweisung *PROC A als erste Zeile in der Prozedur. PROC A protokolliert am Bildschirm alle CFS Prozeduranweisungen, sowie alle an CFS zu übergebenden Datensätze. Werden in einer Anweisung bzw. in einem Datensatz Variable verwendet, so wird der entsprechende Datensatz / die Prozeduranweisung vor und nach der Variablensubstitution am Bildschirm angezeigt.

b) Einschalten des Hardcopy-Modus durch das Kommando HC *datei*,O. *datei* ist der Name der zu verwendenden Hardcopy-Datei. Anhand der Hardcopy-Datei kann der Ersteller einer Prozedur die Folge der an CFS übergebenen Kommandos nachvollziehen und deren Korrektheit überprüfen.

Hinweise zu Prozeduren

Ist der SYSDTA-Eingabesatz kürzer als die erwartete Maskeneingabe, so verhält sich CFS so, als ob die Eingabe am Ende mit ◄ (Endemarke) abgeschlossen worden wäre.

Bei Eingabe von Kommandos über SYSDTA werden die Ausgabeteile der Bildschirmmasken mit WROUT ausgegeben. Dies kann durch /SETSW ON=4 verhindert werden (reduziertes Ausgabevolumen).

Rückkehr in die Grundebene von CFS nach Eröffnung von Connections:

Im prozedurgesteuerten Modus von CFS kann die K2-Taste nicht simuliert werden. Eine Möglichkeit, ohne K2-Taste aus einer Connection in den CFS-Basisprozeß zu gelangen, besteht in der Verwendung des Fluchtsymbols ^.

Beispiel:

```
OC1/LOGON user-id1,acct1,C'passw1'
^
OC2/LOGON user-id2,acct2,C'passw2'
^
```

Wird an irgendeiner Stelle in einem Datensatz einer CFS-Prozedur das Konstrukt &(*name*) angegeben, so wird dieses vom CFS-Eingabeprozessor durch den momentanen Inhalt der Job-/SDF-P Variablen ersetzt. Hierbei kann für *name* auch eine CFS-Variable angegeben werden Die CFS-Variable muß als Wert den Namen der gewünschten Jobvariablen enthalten: &(&CFSVAR) .

Auf diese Weise ist es z.B. möglich, beim automatischen Eröffnen von Connections zu $DIALOG im LOGON-Kommando das Passwort durch den aktuellen Inhalt einer Variablen zu versorgen.

Beispiel:

```
OC1/LOGON user-id1,acct1,&($TSOS.TAGESPASSW)
^
```

In diesem Beispiel enthält die Jobvariable $TSOS.TAGESPASSW als Wert eine Zeichenfolge von der Art C'....' bzw. X'....'.

BS2000-Kommandos, die prozedurgesteuert in einer durch OC eröffneten Connection abgesetzt werden, sollten nicht mit einem Schrägstrich beginnen.

Beispiel:

```
 /STA L
OC1/LOGON user-id,acct-nr,C'passw'
EXEC programm
```

BS2000-Kommandos müssen mit einem führenden Blank angegeben werden. Anderenfalls würde die End-of-File Bedingung für SYSDTA wirksam werden. Dies gilt jedoch nur, falls CFS in einer BS2000 DO-Prozedur aufgerufen wird.

Beispiele für Prozeduren

Mit der im folgenden dargestellten Prozedur (gespeichert im Element L in der CFS.USERLIB) kann der Benutzer bestimmte, am Bildschirm aufgeführte Dateien/Bibliothekselemente mit dem Action-Code X ankreuzen. Die angekreuzten Datenobjekte werden nach Eingabe des Kommandos A (Ausführen) mit der Variablen Action ONXLIST formatiert in eine druckaufbereitete Liste geschrieben und ausgedruckt. Die CFS-Prozedur wird aufgerufen durch DO L bzw. DO L,*datei*. Im letzteren Fall wird der erste Stellungsparameter automatisch der Prozedurvariablen &1 zugewiesen.

```
*IF  &1 = ''  &1 = '#LIST.FRANZ'
*SET TAB=%
ONXLIST &1
*WRITESYS 'Ankreizln,  wosd ausdruckt ham willst'
*ON %TERM CONTINUE
*DIA
U
NPFSTAT &1
%PD7S
*           Ausdruck der erzeugten Liste am Stationsdrucker 7 mit Space=E  (PD7S)
U
*           selektierte Liste wieder anzeigen.
RL
```

Die folgende Prozedur prüft, ob in CFS bereits eine Connection zu der UTM-Anwendung TEST eröffnet ist. Ist dies der Fall, so wird in diese Connection verzweigt. Ansonsten ermittelt die Prozedur die nächste freie Connection, stellt die Verbindung zu TEST her und setzt ein KDCSIGN Kommando ab.

```
*&N=0
.LOOP
*&XX='CON'&N
*IF &&XX = 'TEST' SKIP .CC
*&N=&N+1
*IF &N<10 SKIP .LOOP
*IF &NEXTCON = '' SKIP .NOFREECONN
OC&NEXTCON TEST,CFS1
KDCSIGN ......
*SKIP .ENDE
*
.NOFREECONN
*WRITESYS 'Alle Connections belegt.  Anwendung TEST kann nicht eröffnet werden'
*SKIP .ENDE
*
.CC
CC&N
*
.ENDE
```

In der folgenden BS2000 DO-Prozedur wird in allen Elementen vom Typ J (Job-Control) in einer PLAM-Bibliothek nach dem Vorkommen des Strings '12132716' (LOGON-Paßwort einer Benutzerkennung) gesucht. Die Treffersätze werden in eine gesonderte SAM-Datei geschrieben. In einem folgenden Schritt wird die Trefferdatei mit EDT in der Weise bearbeitet, daß alle Strings '12132716' in '884310' umgewandelt werden. Über das REWRITE-Kommando von CFS werden die ursprünglichen Bibliothekselemente mit dem neuen Inhalt aktualisiert.

```
/PROC C,(&BIBL=LMS.PROC,&PSWALT='12132716',&PSWNEU='884310'),SUBDTA=&
/SYSFILE SYSDTA=(SYSCMD)
/SETSW ON=(1,4)
/EXEC CFS
*002 FILE=LMS &BIBL;FCB=J;VAR=ON&FIND'&PSWALT'=W #TREFFER
AU
NQR
EDT,#TREFFER
ON&:2-2:FIND'*'COPY TO (1) KEEP
ON&CA'&PSWALT'T'&PSWNEU'
COPY &(1)
WRITE O
HALT
REWR #TREFFER
END
/SETSW OFF=(1,4)
/ENDP
```

Die nachfolgend beschriebene DO-Prozedur realisiert eine Konsole, die die folgenden Funktionen ausführt:

1) Systemdumps werden automatisch mit *tsn*.Y beantwortet.
2) Über das Kommando /TYPE CFSCON:/*cmd* bzw. /TYPE CFSCON:*tsn.xxxx* können von Benutzertasks, die unter den Kennungen TSOS oder TT08 angeloggt sind, beliebige Konsoleingaben getätigt werden.
3) Die DO-Prozedur wird durch Eingabe des Kommandos /TYPE TERMINATE CONSOLE beendet.

```
/PROC N
/SYSFILE SYSDTA=(SYSCMD)
/SETSW ON=(1,4)
/EXEC $CFS.CFS
*002 FILE=NO
OC9$CONSOLE,,,R
xxxxx,C'sssss'
*&STAM='Y'
*ON %OUT=($CFS.CFS.PROCLIB(X/SRCITEM)) RESUME
CC9/STA MSG                  nachschauen, ob Anfragen ausstehen
*SKIP .SETTSN
.LOOP
*IF &STAM='N' SKIP .ONOUT
*SEND '/STA MSG'
.ONOUT
*ON %OUT                     letzte *ON %OUT-Bedingung aktivieren
.SETTSN
*&TEXT=%PTR(1,80)            Meldungstext merken
*WRITE '&TEXT'               Meldungstext ausgeben
*&TASK=%PTR(1,4)
*SKIP &SLABEL
.NOMSG
*&STAM='N'
*SKIP .LOOP
.SYSDUMP
*WRITE '-------->  &TASK..Y'
*SEND '&TASK..Y'
*SKIP .LOOP
.CFSCONSOLE
*IF %OUT=:1:'%' &TASK=%PTR(1,4)
*&VAR='USER-ID('&TASK.')'
*IF &&VAR='TSOS' SKIP .ALLOW-CONSINPUT
*IF &&VAR='TT08' SKIP .ALLOW-CONSINPUT
*WRITE '/MESS TSN=&TASK,USER-ID &&VAR. NICHT ZUR KONSOLEINGABE BERECHTIGT'
*SEND '/MESS TSN=&TASK,USER-ID &&VAR. NICHT ZUR KONSOLEINGABE BERECHTIGT'
*SKIP .LOOP
.ALLOW-CONSINPUT
*&TEXT2=&TEXT(1,50)
*WRITE 'KOMMANDO AUSGEFUEHRT: &TEXT2'
*SEND '&TEXT'
*SEND '/MESS TSN=&TASK,KOMMANDO AUSGEFUEHRT: &TEXT2'
*SKIP .LOOP
.EXIT
*ON %TIME=0 RETURN           Rückkehr in das Kommandofeld von CFS
DC9
END                          CFS beenden
/SETSW OFF=(1,4)
/ENDP
```

Die Bedingungen, auf welche Konsolmeldungen reagiert werden soll, sind in dem Element X/SRC ITEM der PLAM-Bibliothek $CFS.CFS.PROCLIB gespeichert. In unserem Beispiel hat das Element den folgenden Inhalt:

```
'SYSTEMDUMP DESIRED'+:1-20:'?' .SYSDUMP
'CFSCON:' .CFSCONSOLE
'NO MESSAGE OUTSTANDING ON THE CONSOLE' .NOMSG
'TERMINATE CFSCONSOLE' .EXIT
```

20. Startup

Der Startup-Mechanismus von CFS bietet dem Benutzer die Möglichkeit, nach dem Programmaufruf automatisch eine Folge von Kommandos auszuführen. Die Kommandos/Selektionseingaben werden aus einer SAM-Datei (Startup-Datei) gelesen. Durch die beim Startup ausgeführten Kommandos können z.B. benutzerindividuelle Programm-Modi gesetzt werden. Die in Startup-Dateien verwendete Syntax ist im Kapitel "Prozedursprache", siehe Seite 485 und folgende ausführlich dargestellt.

Hierarchie der Startup-Dateien

1) Nach dem Programmaufruf sucht CFS als erstes nach einer Datei mit dem **Link-Namen** CFSCMD oder CFSCMD1. Wurde vom Benutzer ein File-Kommando mit diesem Link-Namen abgesetzt, so wird die entsprechende Datei als Startup-Datei verwendet.
   Eine mit Link-Namen CFSCMD zugewiesene Datei wird nur im Dialogbetrieb ausgeführt. Eine mit Link-Namen CFSCMD1 zugewiesene Datei wird auch bei Aufruf von CFS in Prozeduren oder im Enter (SETSW ON=1) ausgeführt. Ausnahme: SETSW ON=11.

2) Existiert kein TFT-Eintrag mit dem Link-Namen CFSCMD, so wird der im LOGON-Kommando angegebene **Job-Name** gesucht (/.name LOGON ...) und an das Prefix 'CFS.STARTUP.' angehängt. Existiert unter der eigenen Benutzerkennung eine Datei mit dem Namen CFS.STARTUP.*name*, so wird diese Datei als Startup-Datei verwendet.

   Existiert keine Datei CFS.STARTUP.*name* unter der eigenen Benutzerkennung, versucht CFS eine zentrale Datei dieses Namens unter der Benutzerkennung zu finden, unter der das Programm CFS eingerichtet wurde (normalerweise $CFS. bzw. $TSOS.).

3) Existiert keine der oben genannten Dateien, so wird eine Datei mit Namen CFS.STARTUP.STD unter der eigenen Benutzerkennung gesucht.

   Existiert die Datei CFS.STARTUP.STD nicht unter der eigenen Kennung, versucht CFS als letztes, eine zentrale Datei dieses Namens unter der Benutzerkennung zu finden, unter der das Programm CFS eingerichtet wurde (normalerweise $CFS. bzw. $TSOS.).

Die Startup-Datei ist eine CFS-Prozedur, die beim Start von CFS automatisch ausgeführt wird. Für die prozedurbezogenen Sprachelemente, die auch in einer Startup-Datei verwendet werden, siehe Seite 485 und folgende (Prozedursprache).

Hinweis:

Falls eine Startup-Datei mit den oben angegebenen Namenskonventionen existiert, der Benutzer diese für den aktuellen Lauf von CFS jedoch nicht verwenden möchte, so ist vor dem Aufruf von CFS das File-Kommando /FILE *DUMMY,LINK=CFSCMD abzusetzen.

Beispiel für eine Startup Datei:

```
*002 FILE=NO
/TCHNG OFLOW=NO
*PROMPT &VAR,'Soll Connection 1 zur Benutzerkennung ..... eröffnet werden ?  (J/N)'
*IF &VAR = 'N'  SKIP .OC2
1/LOGON kennung1,acct,&(PASSW)
^
.OC2
*PROMPT &VAR,'Soll Connection 2 zur Anwendung UTM1  eröffnet werden ?  (J/N)'
*IF &VAR = 'N'  SKIP .NO-UTM
2UTM1,CFSTERM1
KDCSIGN .......
^
.NO-UTM
*WRITE 'Laden der P-Tasten '
PL modul
*WRITE 'Kommandogedächtnis laden'
* Kommandogedächtnis laden                              (Kommentar)
*&SUFFIX = &LOGNAM
*IF &SUFFIX = ''  &SUFFIX = &ACCT
LM CFS.MEM.&SUFFIX
*  Standard User Option definieren                      (Kommentar)
UO VSN
/TCHNG OFLOW=ACK
*
*ON %END CONTINUE
*
*  Die obenstehende ON-Anweisung bewirkt, daß bei Beendigung des Programms CFS
*  durch den Benutzer  (Kommando  END / *), der Prozedurmodus automatisch mit
*  der auf  *DIA  folgenden Anweisung fortgesetzt wird.  (Shutdown-Kommando-
*  folge,  siehe folgende Seite)
*
*  Umschalten der Eingabe auf die Datensichtstation.
*DIA
```

Der oben angegebene Teil der Startup-Datei wird nach dem Aufruf von CFS automatisch ausgeführt. Sobald der *DIA-Satz gelesen wird, schaltet CFS das Eingabemedium auf die Datensichtstation um.

Die Prozedur erhält die Kontrolle erst dann wieder, wenn der Benutzer CFS beendet durch Eingabe der Kommandos END, *, /DO ..., /CALL ..., /LOAD ..., /EXEC ..., /LOGOFF bzw. durch Betätigung der K1-Taste. Die Prozedur wird auch durch Eingabe des Kommandos RET wieder aufgenommen.

Nach der Rückkehr in die Prozedur wird der nachfolgend dargestellte Teil der Startup-Datei ausgeführt.

```
*
*  Kontrollierte Beendigung von CFS  (Shutdown).
*
NPNO
*WRITE 'Endebehandlung'
*PROMPT &SM,'Soll das Kommandogedächtnis gesichert werden ?  (J/N, DUE1 = J)'
*IF &SM = 'N'  SKIP .CONEND
*  Zurückschreiben des Kommandogedächtnisses.
*
*&SUFFIX = &LOGNAM
*IF &SUFFIX = ''  &SUFFIX = &ACCT
*
SM CFS.MEM.&SUFFIX
.CONEND
*&N=0
.LOOP
*&CON='CON'&N
*IF &&CON = '' SKIP .CON1
*                                  (Connection n nicht eröffnet)
*IF &&CON = '$DIALOG'  SKIP .CC1
*                                  (Connection zu $DIALOG)
DC&N
*SKIP .CON1
.CC1
CC&N.B
*                                  (Continue in Conn. &N + Break)
LOGOFF NOSPOOL
.CON1
*&N=&N+1
*IF &N<10 SKIP .LOOP
.ENDE
*  Beenden von CFS
END
```

Das Beispiel zeigt eine Startup-Datei mit Shutdown-Kommandofolge für CFS. Die Shutdown-Kommandofolge besteht aus den Anweisungen nach dem *DIA-Satz.

Ein weiteres Beispiel für eine Startup-Datei finden Sie auf Seite **467**.

21. Userlib

Benutzerspezifische Userlib

Die CFS-Userlib ist eine benutzerspezifische Bibliothek, in der CFS-Prozeduren, Kommandogedächtnisse (durch Save Memory erzeugte Dateien) und Dialogmitschnitte (LOG-Dateien) abgelegt werden.

Standardmäßig hat die Userlib den Namen CFS.USERLIB. In erster Instanz wird somit für jede Benutzerkennung eine eigene Userlib gesucht.

Durch das Kommando UL *bibl* kann der Benutzer eine andere Userlib zuweisen. Die Änderung des Namens der Userlib kann auch in der SET-Maske von CFS erfolgen.

Für die USERLIB sollten PLAM-Bibliotheken verwendet werden. Die einzelnen Prozeduren sind unter dem Typ 'S' (Source) abzuspeichern.

Das Kommando DO *xxxxx* | RET *xxxxx* wird von CFS in folgender Weise abgearbeitet: Falls *bibl*(*element*) bzw. %*name* angegeben wurde, so ist dadurch die auszuführende Prozedur eindeutig bestimmt. In allen anderen Fällen wird gemäß der folgenden Hierarchie nach der Prozedur gesucht:

a) CFS sucht in der zugewiesenen Userlib (Standard: CFS.USERLIB) nach einem Element mit dem Namen S/*xxxxx*.

b) Es wird in der zentralen USERLIB nach einem Element mit dem Namen S/*xxxxx* gesucht. Die zentrale USERLIB kann mit /FILE ...,LINK=USERLIB zugewiesen werden oder wird als Bibliothek mit dem Namen $*user-id*.CFS.USERLIB.STD gesucht. $*user-id* = Benutzerkennung in der die Programmphase CFS gespeichert ist.

c) Es wird eine Datei mit dem Namen *xxxxx* gesucht.

Zentrale Userlib

Für alle CFS-Benutzer verfügbare Prozeduren können in einer besonderen Userlib zur Verfügung gestellt werden. Diese Userlib ist nur einmal im System unter dem Namen $*user-id*.CFS.USERLIB.STD vorhanden. $*user-id* ist dabei die Benutzerkennung, unter der alle zu CFS gehörigen Dateien (z.B. CFS, CFSLIB, CFSHELP) gespeichert sind.

Für den Benutzer besteht jedoch auch die Möglichkeit, per FILE-Kommando eine beliebige andere Userlib zuzuweisen: /FILE *userlib*,LINK=USERLIB.

Userlib und Kommandogedächtnis / Restore

In den Kommandos LM (Load Memory) und RES (Restore Dialog) können ebenfalls Elemente aus der Userlib verwendet werden. Im Gegensatz zu den Kommandos DO und RET wird durch LM *xxxxx* und RES *xxxxx* stets eine Datei dieses Namens angesprochen. Ein Bibliothekselement muß mit der vollständigen Bezeichnung angegeben werden: *userlib*(*xxxx*).

22. Wiederanlauf nach DUMP / Diagnose

Falls CFS auf eine abnormale Programmunterbrechung (PDUMP) läuft, so kann der Benutzer durch das BS2000-Kommando **/INTR R** in den meisten Fällen einen kontrollierten Wiederanlauf des Programms bewirken. Dadurch wird verhindert, daß bestehende Connections, das Kommandogedächtnis von CFS und der Inhalt der EDT-Arbeitsbereiche verloren gehen.

Während der Ausführung eines langdauernden Suche-Kommandos im Display-Modus von CFS sowie bei der Ausführung des Kommandos '++' beim Display einer großen SAM-Datei, kann durch Betätigen der K2-Taste (Verzweigen in die Kommando-Ebene des BS2000) und anschließendes /INTR R die laufende Operation unterbrochen werden. Der Display-Bildschirm wird in diesem Fall auf den zuletzt gelesenen Satz in der Display-Datei positioniert.

Zur Erstellung von Diagnoseunterlagen bei einer abnormalen Programmunterbrechung kann das BS2000-Kommando **/INTR %MAPLST** eingegeben werden. Die daraufhin erscheinende Bildschirmmaske kann mit der F3-Taste in eine Hardcopy-Datei von CFS oder durch Betätigen der LA1-Taste auf einen lokalen Drucker protokolliert werden. Die jeweiligen Ausdrucke können zusammen mit einer Beschreibung der Eingaben, die zu dem Fehler führten, an die Firma OPG gesendet werden (Fax. +49/89/260 9929).

Kontrollierter User-Dump: Mit der K2-Taste in den BS2000-Kommandomodus verzweigen und anschließend **/INTR DUMP** eingeben.

23. Terminierungsabfrage

Nach der Ausführung der Variablen Actions fragt CFS den Benutzer, was er als nächstes tun möchte:

"Term / New / Updated list / U+Keep var / last select ? (T/N/NP../U/UK/UE/*/?)"

Die Benutzereingabe entscheidet über den weiteren Programmablauf:

T — Das Programm CFS wird beendet.

N — CFS bietet dem Benutzer wieder eine leere Selektionsmaske an. Aufgrund der Parametereingaben kann eine neue Dateienliste aufgebaut werden.

NP *param* — Das Feld FILENAME-SELECT und andere Felder der Selektionsmaske werden mit dem in *param* angegebenen Inhalt gefüllt. Anschließend wird die entsprechend den Selektionsangaben aufgebaute Dateienliste angezeigt. Das Ausfüllen der Selektionsmaske wird damit übersprungen.

U — CFS zeigt die letzte Dateienliste in aktualisierter Form an. Alle Veränderungen, die sich aufgrund der zuvor ausgeführten Actions ergeben haben, sind in der upgedateten Dateienliste berücksichtigt. Insbesondere sind die Einträge für gelöschte Datenobjekte aus der Dateienliste entfernt (intern mit dem Action-Code '-' versehen). Der Benutzer kann sich die unsichtbaren Einträge der Dateienliste mit dem Kommando YANK wieder anzeigen lassen. Eine im letzten Durchgang definierte Variable Action gilt für den neuen Verarbeitungsgang nicht mehr.
In der Dateienliste eingetragene Action-Codes X für Variable Actions werden nicht gelöscht.

UK — Analog zur Eingabe U , jedoch wird die Variable Action nicht gelöscht.

UE — Es werden nur die Einträge der letzten Dateienliste angezeigt, für die die Variable Action nicht fehlerfrei ausgeführt werden konnte. Die Einträge der Dateienliste, für die keine Variable Action oder die Variable Action ohne Fehler ausgeführt wurde, werden intern mit dem Action-Code '-' unsichtbar gemacht. Die vollständige Dateienliste kann mit dem Kommando YANK wieder angezeigt werden. Ohne Eingabe von UE werden fehlerhaft ausgeführte Variable Actions in der rechten Fehlerspalte der Dateienliste mit 'ERROR' gekennzeichnet.

* — Es wird in die Selektionsmaske verzweigt, wobei diese bereits mit den zuletzt eingegebenen Selektionskriterien vorbesetzt ist.

? — Aktivierung des Help-Systems. Es werden die verschiedenen Eingabemöglichkeiten beschrieben.

Hinweise:

Die leere Antwort auf die Terminierungsabfrage (Drücken der ENTER-Taste) hat die gleiche Wirkung wie die Eingabe U, d.h. es wird die letzte Dateienliste in aktualisierter Form angezeigt. Im Unterschied zur expliziten Eingabe von U werden jedoch alle im letzten Schritt eingetragenen X Action-Codes auf Blank gelöscht. Bei der Beantwortung der Terminierungsabfrage mit U werden die zuletzt eingetragenen X-Action Codes in der Dateienliste wieder angezeigt.

Bei Beantwortung der Terminierungsabfrage mit ENTER wird der zuletzt angezeigte Ausschnitt der Dateienliste wieder angeboten. Bei Beantwortung der Terminierungsabfrage mit U wird der Anfang der Dateienliste angezeigt.

Die Beantwortung der Terminierungsabfrage kann vorweggenommen werden mit dem Kommando A $[opt]$. Der wahlweise Zusatz opt im Kommando A (Ausführen Actions) gilt als Eingabe für die Terminierungsabfrage. Z.B. werden aufgrund des Kommandos A* die Actions ausgeführt und die Terminierungsabfrage mit * beantwortet. Aufgrund des Kommandos ANP AGE=0 werden die Actions ausgeführt und danach die Terminierungsabfrage mit NP AGE=0 beantwortet. Dies hat die Selektion aller Dateien mit AGE=0 zur Folge.

24. Help-System

Aufruf der Help-Funktion

Der Benutzer von CFS kann zu jedem Zeitpunkt durch Eingabe von ?, ??, ?ALL, ?CON, ?*cmd* oder ?USER in das Help-System verzweigen um sich eine detaillierte Übersicht über die möglichen Eingaben zu verschaffen.

? Es werden alle Informationen zu dem markierten Feld ausgegeben.

?? Es wird eine Anleitung zur Benutzung des HELP-Systems ausgegeben.

?ALL | ?& | HELP Es wird das Hauptmenü des HELP-Systems angezeigt, über das alle vorhandenen HELP-Informationen erreicht werden können.

?NEWS Änderungsprotokoll anzeigen.

?CON Es werden die HELP-Informationen zu den Connections ausgegeben. Diese Zeichenfolge kann auch in Connections, sowohl im Line-Modus als auch im Format-Modus im ersten Feld einer Maske, eingegeben werden.

?*cmd* Es werden alle Informationen zu dem gewünschten Kommando *cmd* ausgegeben.

?USER [, *par*] Anzeige der Menues und Informationen aus der CFSHELP.USER Datei. Für ausführliche Informationen siehe Seite 553.

Beispiele:

COMMAND : `?add`
Es werden alle Informationen zum Kommando ADD ausgegeben.

FILENAME-SELECT : `?fstat`
Es werden alle Informationen zu dem Schlüsselbegriff 'FSTAT' im Feld FILENAME-SELECT ausgegeben.

HELP-Informationen für mehrere Felder einer Maske

Es können in einer Bildschirmmaske auch mehrere Felder gleichzeitig durch '?' markiert werden. Die Erklärungen werden dann der Reihe nach für jedes einzelne Feld angezeigt.

Auswahlmenues des Help-Systems

Wenn die HELP-Informationen sehr umfangreich sind, werden die einzelnen Kapitel in einem Menue angezeigt. Durch Markieren mit einem beliebigen Zeichen können einzelne Menuepunkte ausgewählt werden.

Ein Menue kann sich auch über mehrere Bildschirmseiten erstrecken. In diesem Fall erscheint in der vorletzten Bildschirmzeile der Hinweis "Help continues".

Falls sich hinter einem Menuepunkt ein Untermenue verbirgt, so wird dies durch das Zeichen "ö" angezeigt.

Standardmäßig werden die markierten Menuepunkte einer Bildschirmseite sofort angezeigt. Wenn die Menuepunkte über mehrere Masken hinweg gesammelt und nach Ausgabe der letzten Bildschirmseite des Menues angezeigt werden sollen, so ist die Menuemaske mit der F1-Taste zu senden.

Belegung der Datenübertragungstasten

K1 Die Ausgabe der Informationen zum aktuellen Punkt wird beendet. Die Ausgabe wird beim nächsten Menuepunkt bzw. beim nächsten mit "?" markierten Feld der Maske fortgesetzt. Falls der letzte Menuepunkt bzw. das letzte "?"-Feld der Ausgangsmaske erreicht ist, wird die HELP-Funktion beendet.

K3 Die Ausgabe der Informationen zum aktuellen Punkt wird beendet. Die Ausgabe wird fortgesetzt mit der Anzeige des vorhergehenden Menuepunkts fortgesetzt. Nach der Anzeige des ersten Menuepunkts wird in das übergeordnete Menue zurückverzweigt. Die HELP-Funktion wird beendet, wenn das erste Menue erreicht ist.

F1 Wenn sich ein Menue über mehrere Bildschirmseiten erstreckt, so werden die in der aktuellen Bildschirmseite markierten Menuepunkte bei Senden mit ENTER sofort ausgegeben. Wird das Menue jedoch mit der F1-Taste gesendet, so werden die markierten Menuepunkte gesammelt und nach Absenden der letzten Menueseite bzw. bei der ersten Anwendung von ENTER ausgegeben.

F2 Die F2-Taste hat die gleiche Wirkung wie das Kommando MEN (Verzweigen in das übergeordnete Menue).

F3 Die aktuell angezeigte Bildschirmseite wird in die CFS Hardcopy-Datei geschrieben. Sollen mehrere Seiten hintereinander hardcopymäßig mitprotokolliert werden, so empfiehlt sich hierfür das Kommando HC (Hardcopy-Modus einschalten).

Shift ENTER (DUE2)

Das Kommandogedächtnis wird aktiviert. Das in der Help-Maske zuletzt eingegebene Kommando wird angezeigt bzw. der angegebene String wird im Kommandogedächtnis gesucht und die entsprechende Eingabe angezeigt.

Ausgabemasken des HELP-Systems

Eingabe: **??** (Informationen zum HELP-System)

```
dd.mm.yy   hh:mm:ss     H E L P - S Y S T E M                   TSN: ....
COMMAND :
INPUT   :
                      Beschreibung des HELP-Systems

Markieren Sie die gewuenschten Informationen mit "x".

. Wie kann das HELP-System erreicht werden  ("?", "?all", "?operand", "??")
. Eingaben im Feld  "COMMAND"    (Zeile 2 dieser Maske)
. Eingaben im Feld  "INPUT"      (Zeile 3 dieser Maske)
. Bedeutung der Sendetasten K1, F1, K3, F3 und DUE2 im HELP-System
```

Eingabe: **help** oder **?all** (Hauptmenü des HELP-Systems)

```
dd.mm.yy   hh:mm:ss     H E L P - S Y S T E M                   TSN: ....
COMMAND :
INPUT   :
           Hauptmenue des HELP-Systems ueber alle HELP-Informationen

. Aenderungsprotokoll                    .ÖCFS-Parameter
. Fehlermeldungen, Erweiterungswuensche.ÖHardcopy-Funktion von CFS
.ÖHinweise u. Beispiele für Neuanwender.ÖBehandlung von LMS-Bibliotheken
.ÖSelektionsmaske      (Dateiauswahl)  .ÖBenutzereigene CFS-Kommandos
.ÖUser Options                         . CFS-Startup
xÖVariable Actions      (ONX../ON&..)  . CFS-Userlib
.ÖAction-Codes          (E/P/D/C/R..)  . Wiederanlauf von CFS im Fehlerfall
.ÖAllgemeine CFS-Kommandos             . Terminierungsabfrage von CFS
.ÖDisplay-Kommandos      (CFS-Editor)  . Dateitransfer nach MVS
.ÖNebenprozesse         (Connections)  . Dateitransfer von MVS
.ÖMenuesystem fuer Connections         .ÖSendetasten K1 - K3, F1 - F3 und DUE2
.ÖStarten Enter-Prozesse (Job-Report)  .ÖVon CFS benutzte Prozeßschalter
.ÖFile-Transfer         (openFT/RDAC)  . Von CFS benutzte Link-Namen
.ÖDatei-Informationssystem             . Von CFS angelegte Dateien und Jobvar.
. Verwendung des EDT im CFS            .ÖBeschreibung des HELP-Systems
.ÖCFS-Prozedursprache                  .ÖTabellen  (EBCDIC / EBCDIC <--> ASCII)
```

Das Zeichen "ö" besagt, daß unter diesem Menuepunkt ein weiteres Menue folgt.

Eingabe: **?on** (Information über Variable Actions)

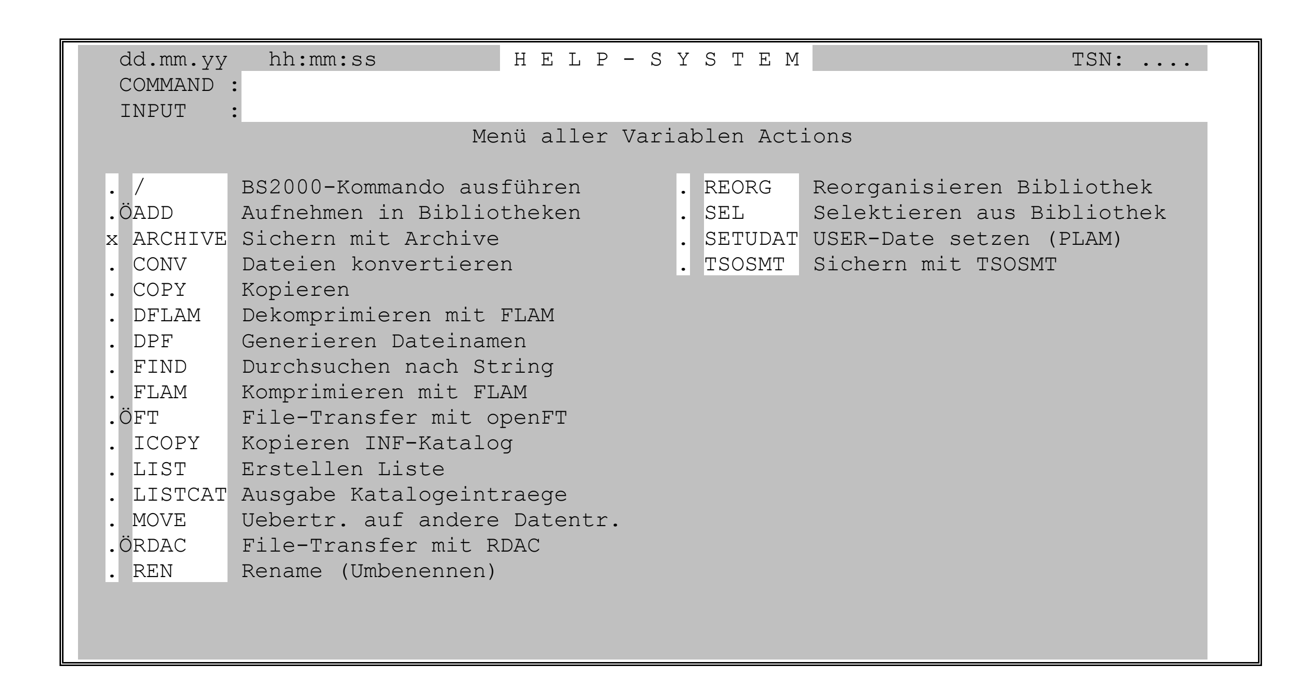
```
   dd.mm.yy   hh:mm:ss              H E L P - S Y S T E M                  TSN: ....
   COMMAND :
   INPUT   :
                            ON      Variable Action definieren

   In diesem Feld werden Aktionen mit umfangreicheren Parametern definiert.
   Die Aktion kann für alle Dateien (&) oder für mit "X" markierte Dateien
   definiert werden.

   Format: ONXvariable Action     Aktion gilt nur für markierte Dateien
           ON&variable Action     Aktion gilt für alle Dateien der Dateienliste


   Bitte wählen Sie:

  xÖMenü aller Variablen Actions
  . Ausführliche Beschreibung des Systems der Variablen Actions
```



```
   dd.mm.yy   hh:mm:ss              H E L P - S Y S T E M                  TSN: ....
   COMMAND :
   INPUT   :
                                Menü aller Variablen Actions

  . /       BS2000-Kommando ausführen        . REORG   Reorganisieren Bibliothek
  .ÖADD     Aufnehmen in Bibliotheken        . SEL     Selektieren aus Bibliothek
  x ARCHIVE Sichern mit Archive              . SETUDAT USER-Date setzen (PLAM)
  . CONV    Dateien konvertieren             . TSOSMT  Sichern mit TSOSMT
  . COPY    Kopieren
  . DFLAM   Dekomprimieren mit FLAM
  . DPF     Generieren Dateinamen
  . FIND    Durchsuchen nach String
  . FLAM    Komprimieren mit FLAM
  .ÖFT      File-Transfer mit openFT
  . ICOPY   Kopieren INF-Katalog
  . LIST    Erstellen Liste
  . LISTCAT Ausgabe Katalogeintraege
  . MOVE    Uebertr. auf andere Datentr.
  .ÖRDAC    File-Transfer mit RDAC
  . REN     Rename (Umbenennen)
```

Eingabemöglichkeiten im Feld COMMAND der Help-Maske

```
dd.mm.yy   hh:mm:ss          H E L P - S Y S T E M                  TSN: ....
COMMAND :
INPUT   :
                        ARCHIVE :  Sichern mit Archive

ONXARCHIVE archive-param

          Es wird eine DO-Prozedur für den Sicherungslauf erstellt. Der Benutzer
          wird gefragt, ob er diese DO-Prozedur im Dialog oder im Enter
          gestartet haben möchte, bzw. keine weiteren Aktionen wünscht
          (nur Generierung einer DO-Prozedur).
          Bei Sicherung im Dialog wird ein DO-Kommando abgesetzt. CFS wird
          damit beendet.


          Beispiele:

          ONXARCHIVE E L=SYSOUT,TAPES=888888

          ONXARCHIVE S NOW=NO,ER=YES,TAPES=123456
```

+ | **-** Eine Bildschirmseite vorwärts / rückwärts blättern.
Falls im Kommandofeld nichts eingegeben wurde, so wird um eine Bildschirmseite vorwärts geblättert.

+*n* | **-***n* *n* Zeilen vorwärts / rückwärts blättern (nicht zulässig in Menues).

++ | **--** Zum Anfang / Ende des aktuellen Punktes (Informationen zu einem Feld oder eines Menuepunktes) positionieren (nicht zulässig in Menues).

LIST [*datei*] [, E|O]

Der gesamte Inhalt des gerade angezeigten Help-Items wird in eine Datei mit dem angegebenen Namen geschrieben. Falls sich der Benutzer z.B. im Änderungsprotokoll von CFS befindet (Kommando ?news), so wird mit LIST CFS.NEWS der gesamte Text des Änderungsprotokolls in eine druckaufbereitete Datei mit dem Namen CFS. NEWS geschrieben.

datei Name der Ausgabedatei. Falls kein Dateiname angegeben wurde, wird als Standardname CFS.HELPLIST.*tsn* verwendet.
Falls bei einem späteren List-Kommando der Dateiname weggelassen wird, werden die Daten an die zuletzt verwendete List-Datei angehängt (Open=Extend).

E Ausgabedatei erweitern (Open=Extend).

O Ausgabedatei wird neu erstellt (Open=Output). Bereits vorhandene Daten werden überschrieben.

DKEY | <u>NDKEY</u> Display Keys. Es wird in der Bildschirmmaske der Isam-Schlüssel des ersten aus der Help-Datei gelesenen Datensatzes angezeigt. Die Information über die Isam-Schlüssel dient nur zu Diagnosezwecken.
Standard: NDKEY

NF Next Field. Informationen zum nächstes Feld ausgeben. Das Kommando entspricht der Bedeutung der K1-Taste.

LF Last Field. Informationen zum vorhergehenden Feld ausgeben (gleiche Wirkung wie K3-Taste).

MEN Zuletzt angezeigtes Menue wieder ausgeben. Das Senden mit der F2-Taste hat die gleiche Wirkung.

S&,.... Stichwortsuche in HELP-Datei.
Es wird in der gesamten HELP-Datei nach dem angegebenen String (z.B. Stichwort Reorganisieren) gesucht. Als Ergebnis wird eine Trefferliste mit allen Menuepunkten ausgegeben, die den Suchstring enthalten.
Für eine ausführliche Beschreibung des Suche-Kommandos siehe Seite **309**.

S,.... String ab der aktuellen Position bis zum Ende der Informationen für das Feld suchen.

Wird das S-Kommando in einer Menuemaske eingegeben, so werden alle Menuepunkte nach dem angegebenen String durchsucht. Die Menuepunkte, in denen der String gefunden wurde, werden mit "S" gekennzeichnet. Ist der Menuepunkt wieder ein Menue, so wird dieser stets mit "S" gekennzeichnet. Das Untermenue wird jedoch nicht durchsucht. Wird die auf diese Weise durch Markierungen vorbelegte Maske gesendet, so werden die Strings in den gekennzeichneten Menuepunkten angezeigt bzw. Untermenues wiederum durchsucht.

S-,.... Suchen String in Richtung Anfang der HELP-Informationen. Dieses Kommando ist nur zulässig nach S,.....

E oder **END** Beenden HELP-Funktion.

/cmd BS2000-Kommando ausführen.

/ Break. Verzweigen in den BS2000-Kommandomodus.

=*arith* Arithmetische Rechnung bzw. Hexadezimal <--> Character Konvertierung.
Ausführliche Beschreibung siehe Seite **212**.

HC | NHC Einschalten / Ausschalten der Hardcopy-Funktion von CFS.
Das HC-Kommando kann mit allen auf Seite **439** dargestellten Parametern angegeben werden. Im Zusammenhang mit dem HELP-System ist die Kommandovariante HC *datei*,R von Interesse (siehe Seite **441**). Es werden damit die am Bildschirm erscheinenden Informationen in reduzierter Form in die angegebene Datei protokolliert. Bei der reduzierten Protokollierung werden die Bildschirmrahmen und die in den Masken konstanten Überschrifts- und Fußzeilen nicht ausgegeben.

? Hinweise zu den Eingabemöglichkeiten im Feld COMMAND der HELP-Maske.

?? Allgemeine Informationen zum CFS HELP-System.

?ALL | **?&** Es wird das Hauptmenü des HELP-Systems angezeigt. Über dieses Menue können alle HELP-Informationen erreicht werden.

MES Mitteilung in CFS-Kommentardatei schreiben. Es wird eine Maske ausgegeben, in der Fehlermeldungen zu CFS, Änderungswünsche und sonstige Mitteilungen an die Fa. OPG eingetragen werden können. Diese Daten werden in die Datei $TSOS. CFSCOM geschrieben. Der Name der Datei kann bei der Installation vom Systemverwalter geändert werden. Zum Format der Kommentarmaske siehe Kommando MES Seite **255**.

Eingabemöglichkeiten im Feld INPUT der Help-Maske

`INPUT :`

Im Feld INPUT der Help-Maske kann der gültige Wert für das Feld der Ausgangsmaske eingegeben und während des Lesens der Help-Informationen vervollständigt werden - selbst über mehrere Bildschirme und Menues hinweg.

Bei Beenden der Anzeige der Help-Informationen durch das Help-Kommando END oder die K1-Taste wird die im Feld INPUT stehende Zeichenfolge als Eingabe in die Ausgangsmaske übertragen.

Benutzereigene HELP-Datei

Aufruf der Benutzermenues

Die Eingabe ?user [,*name*] im Kommandofeld von CFS bewirkt eine temporäre Umschaltung von CFSHELP auf eine vom Benutzer erstellte Helpdatei.

Suchreihenfolge der benutzereigenen Helpdatei

CFS durchläuft bei der Suche nach der zu verwendenden benutzereigenen Helpdatei folgende Schritte:

1) der Name der Helpdatei wurde per File-Kommando zugewiesen (/FILE *xxxx*, LINK=HELPUSER)
2) es existiert eine Datei mit dem Namen CFSHELP.*jobname* bzw. .*acct* falls kein Jobname vorhanden.
3) es existiert eine Datei mit dem Namen CFSHELP.USER
4) es existiert eine Datei mit dem Namen $*userid*.CFSHELP.USER
   userid bezeichnet die Benutzerkennung, unter der das Programm CFS gespeichert ist.

Generierung einer Helpdatei

Folgende Anweisungen sind zur Generierung einer Helpdatei notwendig:

/FILE *helpfile*,LINK=HELPFILE
/EXEC (GENHELP,$CFS.CFSLIB)
DATA=*bibl*(*element-1*)
DATA= ....
......
SOURCE=*bibl*(*element-x*)
END

Die mit *element-n* bezeichneten Bibliothekselemente müssen mit dem Elementtyp S in der PLAM-Bibliothek *bibl* gespeichert sein.

In der Bibliothek CFS.S.LMSLIB ist eine DO-Prozedur J/GENHELP zur Generierung einer User Helpdatei enthalten.

Aufbau der Daten für eine Helpdatei

Die wichtigsten verwendeten Syntaxelemente sind:

:MEN: Menue-Anweisungen definieren Auswahlmenues. Diese können beliebig hierarchisch strukturiert sein.

:HDR: Header-Sätze definieren die Kopfzeilen der Help-Bildschirme

:KEY:/:END: Key- und End-Sätze dienen der Begrenzung der einzelnen Informationseinheiten in Copy-Strecken (siehe unten).
Im SOURCE-Element wird durch Key- und End-Sätze der Beginn und das Ende von Auswahlmenues definiert.

:COP: Cop-Anweisungen bewirken bei der Generierung das Kopieren einer Informationseinheit aus einem DATA-Element.

:DIR: Directory-Anweisung. Alle in der Directory-Liste enthaltenen Menuepunkte können mit dem angegebenen Namen direkt, d.h. durch Eingabe von ?USER,*name* erreicht werden. Beispiel: ?USER,EBCD.
Es ist nur ein Directory-Block pro Helpdatei möglich.

Die Elemente S/CFSHELP.USER.MENUE und S/CFSHELP.HELPTABS in der Bibliothek CFS.S.LMSLIB enthalten als Beispiel alle notwendigen Daten zur Generierung der User Helpdatei.

25. K- / F - / DUE-Tasten in CFS

K1: Allgemeine Rückkehrtaste

	K1		K1		K1	
Display-Modus	--->	Dateienliste	--->	Selektionsmaske	--->	CFS-Ende

Durch **K1** wird ferner der EDT (F-Modus) verlassen, die Anzeige der Help-Texte sowie die weitere Abarbeitung der I/IM-Actions abgebrochen.

Nach dem Aktivieren des Kommandogedächtnisses im CFS-Grundprozeß oder in einer Connection beendet die K1-Taste die weitere Anzeige gespeicherter Kommandos.

Falls der Benutzer in der Menue-Maske (Administration von Connections) die K1-Taste betätigt, so wird dadurch ein Rücksprung nach CFS ausgeführt, sofern das Verfahren "CFS" aufgrund der Menue-Datei für ihn zugelassen wurde.

In einer Connection bewirkt **K1** im Step-Modus des RES-Kommandos die Beendigung des RES-Vorgangs, d.h. es werden keine Eingaben mehr aus der LOG-Datei gelesen.

K2-Taste

Während der Ausführung eines langdauernden Suche-Kommandos kann durch Anwendung der K2-Taste (Verzweigen in die BS2000 Kommando-Ebene) und anschließendes Kommando /INTR R die begonnene Suche abgebrochen werden. Das Sichtfenster wird in diesem Fall auf die zuletzt gelesene Stelle in der Datei positioniert.

Ein mit dem Kommando WAIT angeforderter Wartezustand kann durch Drücken der K2-Taste beendet werden.

In Connections wird über die K2-Taste der Übergang von einer Connection in eine andere bzw. in die Grundebene von CFS gesteuert. Nach Drücken der K2-Taste erscheint die Frage "Return to CFS / Continue in Conn. n / Break ? (R/n/^n/B)".

Im Phantom-Modus des RES-Kommandos hat die Betätigung der K2-Taste die Wirkung, daß die evtl. noch nicht abgelaufene Wartezeit bis zum nächsten Dialogschritt auf 0 gesetzt wird.

K3-Taste

CFS-Editor: Im Query-Modus der Kommandos Suchen mit Ersetzen (S,'...'='...',Q), Suchen mit Wegschreiben (S,'...'=W *datei*,Q), Suchen mit Löschen der Treffersätze (S,'...'=D,Q) wird die Operation nicht ausgeführt und zum nächsten Treffer positioniert. Durch Betätigung der ENTER-Taste wird die Operation ausgeführt und zum nächsten Treffer positioniert.

Bei zeilenweiser Ausgabe des Kommandogedächtnisses bewirkt die Betätigung der K3-Taste das Löschen des angezeigten Eintrags aus dem Gedächtnis. Das Kommandogedächtnis wird um einen Eintrag weiter zurück positioniert.

In der Job-Report-Maske hat die K3-Taste die gleiche Bedeutung wie das Kommando NJR.

Im Test-Compare System (RES ...,C) wird mit der K3-Taste die Anzeige der alten Nachricht und durch weitere Betätigung der K3-Taste die Anzeige der Unterschiede zwischen alter und aktueller Ausgabe gesteuert.

Im Help-System bewirkt die K3-Taste, daß der zuvor angezeigte Abschnitt (Menuepunkt oder Informationen zum vorhergehenden Feld) wieder angezeigt wird.

In der FT-Maske (openFT / RDAC) bewirkt **K3** ähnlich wie im EDT einen vollständigen neuen Bildschirmaufbau (Screen Refresh).

F1-Taste

Im Display-/Modify-Modus von CFS hat **F1** die gleiche Wirkung wie **K1**, nämlich die Rückkehr in die Dateienliste (analog dem Kommando LST). Im Unterschied zur K1-Taste gehen die zuletzt in der Maske eingegebenen Änderungen nicht verloren, sondern werden in die Datei zurückgeschrieben.

Bei unbeabsichtigter Rückkehr aus dem EDT (K1-Taste nach Action-Code EDF*n*) bringt die Funktionstaste **F1** den Benutzer wieder an die zuletzt angezeigte Stelle im EDT zurück. Dies ist allerdings nur solange möglich, als keine neue Dateienliste selektiert wurde.

Im Help-System bewirkt die F1-Taste die Ausgabe der Folgeseite eines Help-Menues.

Im Phantom-Modus des Restore-Vorgangs kann durch Bildschirmlöschen und Eingabe einer Schrittanzahl: +*n*/-*n*/++/-- links oben am Bildschirm und Betätigung der F1-Taste der Dialog um *n* Schritte vorwärts/rückwärts positioniert werden.

F2-Taste

Bei der Anzeige der Dateienliste werden mit **F2** alle angezeigten Dateinamen zum Zwecke des Umbenennens (Rename) überschreibbar gemacht.

Im Display-Modus wird durch **F2** ein implizites Modify-Kommando abgesetzt und der Dateiinhalt wird zum Modifizieren freigegeben.

Im Step-Modus des RES-Kommandos hat die F2-Taste die Wirkung, daß in der LOG-Datei nicht weitergelesen wird, d.h. der interne Satzzeiger bleibt auf dem gleichen Satz stehen. Im nächsten RES-Schritt wird demnach wieder der gleiche Satz angezeigt. Damit kann der Benutzer zusätzliche Eingaben in den RES-Ablauf einfügen.

Im HELP-System hat die F2-Taste die gleiche Wirkung wie das Kommando MEN (Ausgabe des zuletzt angezeigten HELP-Menüs).

F3-Taste

In der Grundebene von CFS können Masken durch Betätigung der F3-Taste in einer Hardcopy-Datei mitprotokolliert werden. Zur Protokollierung von Masken in Connections mit einer Funktionstaste siehe Kommando CH (Connection Hardcopy) Seite 470.
Es ist zu beachten, daß bei Hardcopy mit der F3-Taste, bzw. mit der durch das Kommando CH definierten Datenübertragungstaste jede Einzelmaske auf einer neuen Seite erscheint. Im normalen Hard-

copy-Modus (HC-Kommando) werden nach Möglichkeit mehrere Masken auf einer Druckseite untergebracht.
Das Absenden einer CFS-Maske mit **F3** wird, abgesehen von der Protokollierung, wie ENTER behandelt. Beim Hardcopy einer Maske in einer Connection mit der durch das Kommando CH definierten Datenübertragungstaste gehen die Eingabedaten verloren und müssen noch einmal eingegeben und mit ENTER abgesendet werden.

Im Step-Modus des RES-Kommandos bewirkt die Taste **F3** einen Übergang in den Batch-Modus (d.h. in den automatischen Modus) des Restore.

Shift ENTER (DUE2): Zurückblättern im Kommandogedächtnis

CFS führt intern Tabellen, in denen alle eingegebenen Kommandos (Feld COMMAND in der zweiten CFS-Maske), sowie alle Eingaben im Feld FILENAME-SELECT der Selektionsmaske gespeichert werden. Auf diese Gedächtnisse kann der Benutzer mit der Taste Shift ENTER (DUE2) zugreifen (sequentiell durch ⊔ Shift ENTER bzw. assoziativ durch *string* Shift ENTER). Das Zurückblättern geschieht durch wiederholtes Betätigen der Taste Shift ENTER.
Der Connection-Handler von CFS führt für jede Connection ebenfalls eine Tabelle mit den getätigten Eingaben (Line-Mode). Dieses Gedächtnis wird aktiviert durch Eingabe von -◄ ENTER bzw. -*string*◄ ENTER. Das weitere Zurückgehen im Kommandogedächtnis erfolgt dann durch wiederholte Betätigung der Taste Shift ENTER.

Bestimmte Connection-Switch Systeme wie z.B. SIMBA benutzen die Taste Shift ENTER (DUE2) zur Rückkehr in das Grundmenue. Falls CFS unter einem dieser Systeme läuft, kann das Kommandogedächtnis von CFS mit der Taste Shift ENTER nicht aktiviert werden. In diesem Fall kann CFS angewiesen werden, eine andere Datenübertragungstaste zur Aktivierung des Kommandogedächtnisses zu verwenden. Mehr hierzu finden Sie im Quellprogramm des Initialisierungsmoduls CFSMAIN unter dem Stichwort "Alternative Taste". Der Tastencode für das Kommandogedächtnis kann auch über die Startup-Datei modifiziert werden. Um das Kommandogedächtnis mit der F5-Taste zu aktivieren, ist z.B. folgende Anweisung als Kommando in die Startup-Datei aufzunehmen: M L'2545',X'5F'

In einer Connection zu einer UTM-Anwendung sind asynchrone UTM-Administrationsnachrichten (freilaufende Ausgaben) durch Betätigung der Taste Shift ENTER (DUE2) zu quittieren. Shift ENTER darf nicht für Verarbeitungsfunktionen der UTM-Anwendung benutzt werden. Falls dies dennoch gewünscht ist, so muß die Datenübertragungstaste für freilaufende Ausgaben umdefiniert werden durch das Kommando M L'2307',X'*nn*'. X'*nn*' ist der ASCII-Code für die neu festzulegende Datenübertragungstaste. Durch das Kommando M L'2307',X'3A' wird beispielsweise die Shift ENTER-Funktion auf die Taste **K14** gelegt. Näheres zu den Codes für die Datenübertragungstasten finden Sie im Manual "TRANSDATA, Datensichtstation 9750, Schnittstelle für Programmierer".

ENTER (DUE1): gewöhnliche Datenübertragung

Alle CFS-Masken werden normalerweise mit der ENTER-Taste übertragen.
Die Verwendung einer Endemarke (◄) innerhalb einer Maske hat zur Folge, daß alle Zeichen rechts und unterhalb der Endemarke nicht übertragen werden. CFS nimmt für diese nicht übertragenen Daten Blanks als Standardwert an.

Beispiel:

Das Löschen aller Action-Codes in einer Maske erfolgt durch Eingabe von -0◄ ENTER im Kommandofeld. Datenänderungen im Modify-Modus werden nicht ausgeführt, falls im Kommandofeld von CFS die Endemarke eingetragen wurde. Falls im Modify-Bereich der CFS-Maske die Endemarke angegeben wurde, so wird beim Zurückschreiben der Daten nur der Teil des Bildschirms links und oberhalb der Endemarke berücksichtigt. Änderungen im übrigen Teil des Bildschirms werden nicht in die Datei zurückgeschrieben.

26. Prozeßschalter für besondere Betriebsmodi

`SETSW ON=1` CFS erwartet die Eingaben von SYSDTA. Als Eingabeaufforderung erscheint am Bildschirm ein geschützter Stern (*).

`SETSW ON=2` Alternativer Maskenmodus von CFS. Dieser Modus ist z.B. notwendig beim Betrieb von CFS auf PC-Emulationen).
CFS löscht den Bildschirm vor jeder Maskenausgabe. Die logische Zeilenlänge wird fest auf 80 Stellen gesetzt (notwendig beim Betrieb von CFS an Datensichtgeräten von graphischen Systemen).
Falls CFS als Unterprogramm eines anderen Hauptprogramms aufgerufen wird, so bewirkt der Schalter 2, daß CFS keinen ESCPBRK-Stxit aktiviert außer beim Eröffnen von Connections.

`SETSW ON=4` Der Umfang der von CFS produzierten Bildschirmausgaben wird reduziert:

- Bei CFS in DO-/Enter-Prozeduren (Variante *002) werden die WROUT-Teile der Masken am Bildschirm/im SYSOUT-Protokoll nicht ausgegeben.
- Im prozedurgesteuerten CFS-Modus werden die *001-/*002-Sätze am Bildschirm/im SYSOUT-Protokoll nicht wiedergegeben.
- Im Connection-Handler wird nach Betätigung der K2-Taste die Ausgabe des Connection-Display (Übersicht aller Connections) unterdrückt.
- Beim Wiederablaufenlassen eines mitgeschnittenen Dialogs (RES-Kommando) werden die automatisch getätigten Eingaben am Bildschirm nicht abgebildet.

`SETSW ON=10` Bewirkt die Einstellung des NAC-Modus (No Add Compress) bei ADD-Operationen für LMS-ISAM Bibliotheken. Beschreibung siehe Kommando NAC Seite **467**.

`SETSW ON=11` Dieser Prozeßschalter muß gesetzt werden, falls CFS in einer DO-/Enter-Prozedur abläuft und *001-Sätze verwendet werden (Basismodus).

`SETSW ON=12` Bewirkt die Einstellung des AI-Modus (Add with Increment of version) bei LMS ADD-Operationen. Das Einschalten des AI-Modus mit Hilfe des Prozeßschalters 12 wirkt nur im Basismodus von CFS-Prozeduren (/SETSW ON=11, *001 ...). Beschreibung siehe Kommando AI Seite **468**.

`SETSW ON=14` Unterdrückung der Ausgabe der Dateinamen in Variablen Actions. Nach Rückkehr in die Dateienliste mit der Antwort U auf die Terminierungsabfrage werden nur noch die Dateien angezeigt, für die die vorausgegangene Variable Action nicht erfolgreich ausgeführt wurde. Dateien, bei denen die Variable Action erfolgreich war, werden von CFS durch Eintrag des Action-Codes "-" aus der Dateienliste entfernt.

`SETSW ON|OFF=31` Dieser Schalter sollte vor jedem Aufruf von CFS innerhalb einer Prozedur auf OFF gesetzt sein. Werden von CFS keine den Selektionsbedingungen entsprechenden Datenobjekte gefunden, so wird der Schalter 31 auf ON gesetzt. Damit kann in DO-/Enter-Prozeduren sowie in CFS-Prozeduren auf einfache Weise geprüft werden, ob von CFS Dateien mit den gesuchten Merkmalen gefunden wurden oder nicht. Der Prozeßschalter 31 wird von CFS auch gesetzt, falls bei der Ausführung der Actions ON&COPY/ ON&ADD/ ON&SEL Fehler aufgetreten sind bzw. falls bei den Kommandos SI/SIL/SIS keine Termine der gewünschten Art gefunden wurden.

Hinweis:

Bei Beendigung von CFS im Prozedurmodus werden die **Prozeßschalter 1 und 11 zurückgesetzt**. Bei einem erneuten Aufruf von CFS in der gleichen Prozedur sind die Schalter daher wieder neu zu setzen.

27. Linknamen

Im folgenden werden die für Benutzer relevanten Link-Namen von CFS beschrieben.

`C1` Eingabedatei-1 für die File-Compare Funktion. Mit /FILE *datei*,LINK=C1 kann der Benutzer die erste Eingabedatei zuweisen. Im Kommando COMP kann dann anstelle des Dateinamens der Linkname C1 angegeben werden.

`C2` Eingabedatei-2 für die File-Compare Funktion. Mit /FILE *datei*,LINK=C2 kann der Benutzer die zweite Eingabedatei zuweisen. Im Kommando COMP kann dann anstelle des Dateinamens der Linkname C2 angegeben werden.

`CEXITnU` benutzereigener Nachrichten-Ein-/Ausgabemodul für Connection *n*.
Mit diesen Linknamen kann eine Modulbibliothek verknüpft werden, aus der der Connection-Handler von CFS einen benutzereigenen Connection-Exit nachlädt. 0 <= *n* <= 9 (Connection-Nummer). Mit /FILE *oml*,LINK=CEXIT*n*U kann der Benutzer seinen eigenen Nachrichten-Ein-/Ausgabemodul ansprechen. Der Standard-CEXIT von CFS wird in diesem Fall nicht aufgerufen. Die LOG- und RES-Einrichtung des Connection-Handlers, sowie das Kommandogedächtnis sind nicht ansprechbar bzw. müssen vom Benutzer in seinem CEXIT selbst programmiert werden.

`CEXITnA` dem Standard-CEXIT von CFS nachgeschalteter benutzerspezifischer Exit.
Mit /FILE *oml*,LINK=CEXIT*n*A kann der Benutzer seinen eigenen Nachbearbeitungs-Modul ansprechen. Der Standard-CEXIT von CFS ruft den CEXIT-A Modul nach erfolgter Nachrichten-Eingabe auf. Dieses Verfahren hat den Vorteil, daß alle Connection-Services wie LOG, RES und Kommandogedächtnis für den Benutzer zur Verfügung stehen. In der CFS.S.LMSLIB findet sich ein Beispiel für einen CEXIT-A, der in den eingegebenen Nachrichten alle Kleinbuchstaben in Großbuchstaben umsetzt.

`CFSCMD` Vom Benutzer per File-Kommando zugewiesene Startup-Datei.
`CFSCMD1` Wie CFSCMD, jedoch wird die Startup-Datei auch bei Aufruf von CFS in Prozeduren und im Enter ausgeführt (SETSW ON=1).

`CFSDISPT` Vom Benutzer per File-Kommando zugewiesene optionale Umsetzdatei für den CFS-Display/ Editor. Falls keine Datei mit Linknamen CFSDISPT zugewiesen wurde, so wird in der Installationskennung von CFS nach einer Datei mit dem Namen CFS. DISPTRTAB gesucht. Falls auch diese Datei nicht existiert, so verwendet CFS die interne Standard-Umsetztabelle.
Die CFS.DISPTRTAB legt fest, welche Zeichen am Bildschirm im Klartext bzw. als Schmierzeichen dargestellt werden. Falls aufgrund des im Rechenzentrum eingesetzten CCS (Coded-Character-Set) Umlaute im CFS-Editor nicht als solche dargestellt werden, so kann der Systemverwalter die entsprechenden Codepositionen in der Umsetztabelle korrigieren. Zwei Musterdateien, mit zusätzlichen Kommentaren versehen, sind unter dem Namen X/CFS.DISPTRTAB (entspricht der Standard-Umsetztabelle von CFS) und X/CFS.DISPTRTAB.EDF041 (für CCS=EDF041) in der ausgelieferten PLAM-Bibliothek CFS.S.LMSLIB abgelegt.

`CFSREORG`

Falls sich die Hilfsdatei beim Reorganisieren von Dateien/Bibliotheken über mehrere Privatplatten erstrecken soll oder falls sie auf einem anderen speziellen Speichermedium eingerichtet werden soll, so kann dies auf die folgende Weise erreicht werden:
Der Benutzer legt auf den gewünschten Datenträgern eine Datei mit beliebigem Namen (Ausnahme: CFS.REORGFILE) und in ausreichender Größe an. Diese Datei wird CFS durch das Kommando /FILE *datei*,LINK=CFSREORG bekannt gemacht. In der Folge werden alle Reorganisationen mit dieser Datei als Hilfsdatei durchgeführt. Die Hilfsdatei wird nach Abschluß der Reorganisation nicht gelöscht, wie dies normalerweise der Fall ist. Es wird lediglich ein Kommando /ERASE *datei*,DATA zum Löschen der Dateninhalte ausgeführt. Der reservierte Speicherplatz bleibt hierdurch bestehen.

`COMPLIST`

Vom Benutzer per File-Kommando zugewiesener Name für das Vergleichsprotokoll der File-Compare Funktion von CFS.

`DISP`

Display-Datei. Mit dem File-Kommando /FILE *datei*,LINK=DISP,FCBTYPE=SAM, STATE=FOREIGN,RECFORM=..,RECSIZE=..,BLKSIZE=.. können auch Banddateien im Display-Modus von CFS angezeigt werden.

`EDTSAM`

Beim Einlesen von Nicht-Standard EDT-Dateien setzt CFS selbständig ein File-Kommando mit diesem Link-Namen ab. Falls mehrere Dateien in verschiedene Ebenen des EDT eingelesen werden sollen, so ist der LINK-Name EDTSAM störend und kann vom Benutzer durch das Kommando /REL EDTSAM entfernt werden.

Über ein File-Kommando mit LINK=HARDCOPY kann eine Hardcopy-Datei mit einem vorgegebenen Namen zugewiesen werden. Dies ist besonders bei der Betätigung der F3-Taste zum Hardcopy einzelner CFS-Masken von Bedeutung, da hier ansonsten in eine Standard-Hardcopy-Datei mit dem Namen CFS.HARDCOPY.xxx protokolliert wird.

`HELPUSER`

Link-Name einer benutzereigenen Helpdatei (siehe Seite **553**).

`IFORMAT`

Formatkatalog des Dateiinformationssystems.
Normalerweise hat der Formatkatalog den Namen $TSOS.CFS.INF.FORMAT. Der Benutzer kann jedoch mit LINK=IFORMAT einen eigenen Katalog zuweisen.

`IFREE`

Feiertagsdatei für Terminverwaltungssystem. Näheres siehe Seite **429**.

`INFCB / OUTFCB`

Link-Namen der Ein- und Ausgabedatei für das FC-Kommando von CFS zur Dateiumsetzung.

`INFO`

INF-Katalog des Dateiinformationssystems.
Normalerweise hat der INF-Katalog den Namen CFS.INF / CFS.INF.JV / CFS.*bibl*. INF / CFS.INF.JES. Der Benutzer kann jedoch mit LINK=INFO einen INF-Katalog zuweisen, der nicht den obigen Namenskonventionen genügt.

`JRSAVE` Vom Benutzer per File-Kommando zugewiesene Datei für Parameter-Sets (SP %*name*). Parameter-Sets können komplexe Suche- und Ersetzungskommandos, Parameterbelegungen für DO-Prozeduren, sowie komplexe Spaltendefinitionen des AD-Kommandos enthalten. Die Aktivierung der Parameter-Sets erfolgt durch die Kommandos S,%*name* / DO %*name* / E %*name* / AD %*name*. Falls kein FILE-Kommando mit LINK=JRSAVE angegeben wurde, so wird die Datei CFS.JRSAVE... verwendet. Die genaue Regel, nach der sich der Name der Standard JRSAVE-Datei bildet, wird einige Seiten weiter unten beschrieben "von CFS angelegte Dateien/Jobvariablen".
Eine Übersicht aller möglichen Parametersets kann durch Angabe von %? angefordert werden. Durch Ankreuzen kann einer der vorgeschlagenen Parametersets ausgewählt werden.

`MAPLIB` Nachladebibliothek für das Kommando PL (P-Tasten laden) und die in der Menue-Datei angegebenen P-Tastenbelegungen (siehe Seite **382** "Menue-System für Connections"). Falls keine Bibliothek mit dem Link-Namen MAPLIB verknüpft ist, werden die Tastenbelegungen aus der PLUS.MAPLIB geladen.

`MENUE` vom Benutzer per File-Kommando zugewiesene Menue-Datei für die Connection-Maske (siehe Seite **381** "Menue-System für Connections").

`MVSIMP` Einspielen von MVS-Bibliothekselementen in eine BS2000 LMS-Bibliothek.
Im Betriebssystem MVS kann mit dem Dienstprogramm IEBUPDTE der Inhalt einer PO (Partitioned Organized) Bibliothek auf Magnetband übertragen werden. Der Inhalt dieser Banddatei kann im BS2000 mit dem Kommando (MVSIMP) in eine LMS-Bibliothek mit dem festen Namen LMS.MVSIMPORT übertragen werden. Vor Eingabe des Kommandos (MVSIMP) muß die Banddatei durch /FILE *datei*,LINK= MVSIMP im BS2000 bekannt gemacht werden. Für weitere Informationen siehe Handbuch für Systemverwalter, "Übernahme von Bibliotheken von MVS nach BS2000".

`PDFILE` Vom Benutzer per File-Kommando zugewiesene PDFILE für besondere Print-Parameter. Falls kein File-Kommando mit diesem Link-Namen angegeben wurde, so wird beim Action-Code PD die vom Systemverwalter eingerichtete zentrale CFS. PDFILE verwendet.

`USERACT` Vom Benutzer per File-Kommando zugewiesene USERACT-Datei für eigene Action-Codes. Die Hierarchie, nach der bei Fehlen des Linknamens der Name der Standard USERACT-Datei ermittelt wird, ist auf Seite **180** beschrieben.

`USERLIB` Vom Benutzer per File-Kommando zugewiesene Userlib. Falls kein File-Kommando mit diesem Link-Namen angegeben wurde, so wird als sekundäre Userlib eine Bibliothek mit dem Namen $*user-id*.CFS.USERLIB.STD gesucht. $*user-id* ist hierbei die Benutzerkennung, unter der alle zu CFS gehörigen Dateien (z.B. CFS, CFSLIB, CFSHELP) gespeichert sind.

von CFS angelegte Dateien und Jobvariablen

`#CFS.SAVELIST.xxxx`

Vom Benutzer durch das Kommando SL *mn* weggeschriebene bzw. von CFS selbständig weggeschriebene Dateienlisten. Diese Dateienlisten können durch das Kommando RL [*mn*] wieder angezeigt werden. Das feste Prefix #CFS.SAVELIST. kann vom Systemverwalter abgeändert werden.

`#Zelement`

Falls ein Bibliothekselement mit dem Action-Code D (Display) markiert wird, so erzeugt CFS eine temporäre Zwischendatei mit dem aufgeführten Namen. Die Zwischendatei wird nach Beendigung des Display wieder gelöscht.

`#datei`

Falls eine Datei bei eingeschalteter ERT-Option (Erase with Retain of Tempfiles) mit dem Action-Code E gelöscht wird, so wird diese Datei von CFS in eine temporäre Datei mit dem Namen *#datei* umkatalogisiert.
Wird ein Element einer Bibliothek mit E gelöscht, so erzeugt CFS eine temporäre Bibliothek mit dem Namen *#bibliothek* und überträgt das Element in diese Bibliothek, bevor es in der Originalbibliothek gelöscht wird.
Die temporären Dateien werden bei /LOGOFF automatisch gelöscht.

`$TSOS.CFSCOM`

Diese Datei wird beim erstmaligen Laden von CFS unter der Kennung TSOS angelegt, mit einem Lesepaßwort versehen und shareable gemacht. In der Datei CFSCOM werden die Erweiterungswünsche und Fehlermeldungen zu CFS abgelegt, die der Benutzer in der durch das Kommando MES angeforderten Bildschirmmaske eingetragen hat.

`CFS.COMPLIST.tsn.yymmdd.hhmmss`

Beim Vergleich zweier Dateien/Bibliothekselemente erzeugtes Vergleichsprotokoll (Kommando COMP).

`CFS.DO.ARCHIVE.tsn`

Aufgrund der Variablen Action ONXARCHIVE erzeugte DO-Prozedur. Diese DO-Prozedur kann im Dialog oder Enter zum Ablauf gebracht werden. Bei Ablauf der Prozedur im Enter wird von CFS zusätzlich eine Enter-Datei mit dem Namen CFS.ENT.ARCHIVE. *tsn* erzeugt.

`CFS.DO.tsn.hhmmss`

Von der Variablen Action ONXDO erzeugte Hilfsprozedur, in der die CALL-Aufrufe der Verarbeitungsprozedur mit den Dateinamen als Parametern enthalten sind. Die erzeugte Hilfsprozedur kann von CFS nicht automatisch gelöscht werden. Das Löschen der DO-Prozeduren mit Namen von der oben angegebenen Art liegt in der Verantwortung des Benutzers.

`CFS.E.tsn.hhmmss`

Vom Job-Report System erzeugte Enter-Datei, falls eine DO-Prozedur mit dem Kommando E im Enter ablaufen soll. Bei Beendigung des Enter-Auftrags wird diese Datei normalerweise gelöscht.

`CFS.EDTSAV [.jobname].tsn.yymmdd.hhmm.x`

Bei Eintritt des STXIT-Ereignisses ABEND, z.B. bei Leitungsverlust, prüft CFS, ob vom Benutzer irgendwann zuvor mit dem Action-Code EDT/EDF/EDL Dateien / Bibliothekselemente bearbeitet wurden. Ist dies der Fall, so werden alle zum ABEND-Zeitpunkt noch belegten EDT-Ebenen in die oben genannten Sicherungsdateien gerettet.

x: Ziffer zwischen 0 und 9, welche die gesicherte EDT-Ebene bezeichnet.

Falls beim LOGON ein Name angegeben wurde, so erscheint dieser ebenfalls in den Namen der erzeugten EDTSAVE-Dateien.

`CFS.HARDCOPY.jobname.yymmdd.hhmm.tsn`

Aufgrund des HC-Kommandos im Grundprozeß von CFS angelegte Hardcopy-Datei, in der die in einer Connection oder in der CFS-Grundebene ausgegebenen Bildschirmmasken zeilen- und spaltengerecht protokolliert werden. Falls im HC-Kommando kein Dateiname angegeben wurde, so wird von CFS der aufgeführte Standardname verwendet.

`CFS.JR.SYSOUT.tsn.hhmmss | CFS.JR.SYSLST.tsn.hhmmss`

Vom Job-Report System angelegte SYSOUT/SYSLST-Datei, falls eine DO-Prozedur über das E-Kommando im Enter zum Ablauf gebracht werden soll. Diese Datei wird von CFS gelöscht, sobald der Benutzer, nach Beendigung des Enter-Auftrags, die Job-Report-Maske mit dem SYSOUT-Listing mit dem Kommando E (End) bzw. mit der K1-Taste quittiert hat.

`CFS.JR.xxxxx.yymmdd.hhmmss`

Vom Job-Report System angelegte **Monitor-Jobvariable** für den zu überwachenden Enter-Auftrag. Diese Jobvariable wird von CFS gelöscht, sobald der Benutzer nach Beendigung des Enter-Auftrags die Job-Report-Maske mit dem SYSOUT-Listing des Enter-Auftrags mit dem Kommando E (End) bzw. mit der K1-Taste quittiert hat. Der Namensteil xxxxx steht für die Job-Report Identification, unter der der Enter-Auftrag gestartet wurde. Näheres hierzu siehe Seite **413**.

`CFS.J2.xxxxx.yymmdd.hhmmss`

Vom Job-Report System wird eine **Jobvariable** dieser Art angelegt, falls in der Prozedurdatei ein /REMARK JRIC-Kommando vorkommt. Diese Jobvariable wird von CFS gelöscht, sobald der Benutzer nach Beendigung des Enter-Auftrags die Job-Report-Maske mit dem SYSOUT-Listing des abgelaufenen Enter-Auftrags mit dem Kommando E (End) bzw. mit der K1-Taste quittiert hat. Der Namensteil xxxxx steht für die Job-Report Identification, unter der der Enter-Auftrag gestartet wurde. Näheres hierzu siehe Seite **413**.

`CFS.JRSAVE....` ISAM-Datei, in der die durch das Kommando SP %*name* gesicherten Parameterbelegungen gespeichert sind. Der genaue Name der verwendeten JRSAVE-Datei wird nach folgender Regel ermittelt:

a) Wurde vom Benutzer ein Kommando /FILE ... ,LINK=JRSAVE gegeben ?

b) Existiert eine Datei mit dem Namen CFS.JRSAVE ?

c) Wurde beim Einloggen des Benutzers ein Jobname angegeben <u>und</u> existiert eine Datei mit dem Namen CFS.JRSAVE.*jobname* ?

d) Falls beim LOGON kein Jobname angegeben wurde und die Bedingungen a) bis c) nicht erfüllt sind, so ergibt sich der Name der JRSAVE-Datei als CFS. JRSAVE.*acct-#*.

Aufgrund der als erstes zutreffenden Bedingung wird der Name der JRSAVE-Datei bestimmt. Durch Modifikation eines CFS-internen Speicherbereichs kann der Name der Standard JRSAVE-Datei vom Benutzer auch frei vergeben werden. Siehe hierzu Seite 479.

Der Benutzer kann in allen Kommandos, die einen Parameterset ansprechen, die JRSAVE-Datei auch direkt angeben: *%name, jrsave*.

`CFS.LOG.yymmdd.hhmm.tsn`
Aufgrund des Kommandos LOG angelegte Logging-Datei, in der die Nachrichten Ein-/Ausgaben für eine Connection mitgeschrieben werden. Die LOG-Datei wird stets im Grundprozeß von CFS angelegt. Der aufgeführte Name der Logging-Datei wird verwendet, falls das Kommando LOG in der Form LOG STD eingegeben wurde.

`CFS.MEM.STD.` .... Aufgrund des Kommandos SM (Save Memory) wird der Inhalt des CFS-internen Kommandogedächtnisses in einer Datei dieses Namens gespeichert. Falls im SM-Kommando kein Dateiname angegeben wurde, so erhält die Datei den Namen CFS.MEM.STD.*jobname*. *jobname* ist der im LOGON-Kommando angegebene Jobname. Falls kein Jobname angegeben wurde, so erhält die Memory-Datei den Namen CFS.MEM.STD.*acct* (*acct*: Abrechnungsnummer der Benutzerkennung).

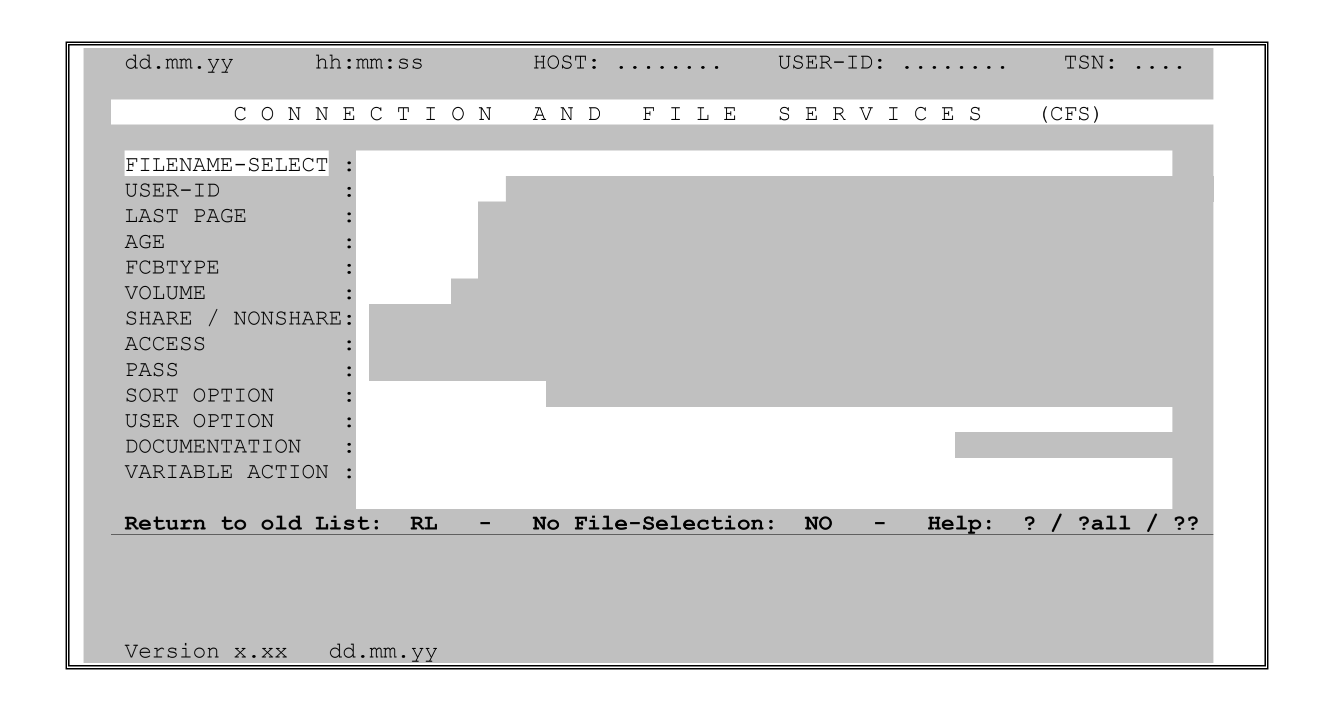
Connection & File Services

Handbuch für Systemverwalter

Ausgabe Januar 2015

Handbuch für den Systemverwalter

Selektionsfeld : FILENAME-SELECT



Alle mit FT zu übertragenden Dateien selektieren (NSTAT)

NSTAT [*ALL | *A | *owner*] [, *remsys*]

Es werden alle Dateien/PLAM-Bibliothekselemente selektiert, die mit openFT zu oder vom eigenen Host übertragen werden und deren Übertragung noch nicht beendet ist.

Falls keine der Optionen *ALL oder *owner* angegeben wurden, so werden alle FT-Aufträge selektiert, für die der Benutzer unter der aktuellen User-Id der Eigentümer ist. Für den Systemverwalter ist dies z.B. die Kennung TSOS.

*ALL — Es werden alle FT-Aufträge selektiert, die von einem beliebigen Benutzer gestartet wurden und noch nicht beendet sind. Die Dateinamen werden vollqualifiziert mit der Benutzerkennung angezeigt.

*A — Wie *ALL, jedoch werden die Dateinamen ohne Benutzerkennung angezeigt.

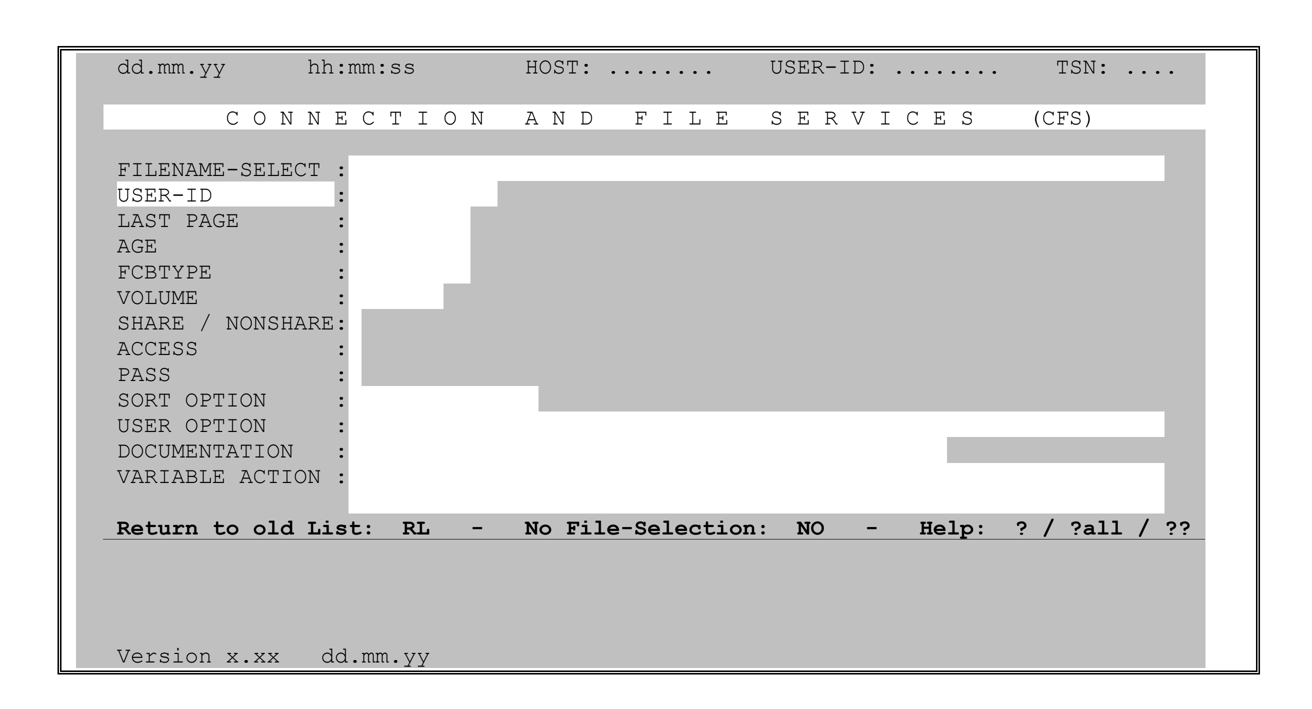
owner — Es werden alle FT-Aufträge selektiert, die von dem angegebenen Benutzer gestartet wurden und noch nicht beendet sind. Für *owner* kann jede beliebige Owner-Identifikation angegeben werden. Eine Owner-Identifikation ist eine Benutzerkennung ohne $ und Punkt, z.B. TSOS .

remsys — Es werden alle FT-Aufträge selektiert, die mit dem angegebenen Partnersystem durchgeführt werden.

Hinweis:

Zu den erweiterten Möglichkeiten des File-Transfers unter TSOS siehe Seite A**591**.

Selektionsfeld : USER-ID


```
dd.mm.yy      hh:mm:ss        HOST: ........    USER-ID: ........    TSN: ....

        C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -   No File-Selection:  NO   -   Help:  ? / ?all / ??

Version x.xx   dd.mm.yy
```

Auswahl von Dateien / Jobvariablen über mehrere/alle Kennungen

(JOIN) | () | [:*x*:|:*[*]:] ( )[-]*string*

(JOIN) Es werden die Default Cat-ID's aller Benutzerkennungen nach Dateien/Jobvariablen mit den gewünschten Eigenschaften durchsucht. Anstelle von (JOIN) kann auch die Abkürzung ( ) angegeben werden.

:x:(JOIN) Es wird der angegebene Pubset *x* auf allen Benutzerkennungen nach Dateien mit den gewünschten Eigenschaften durchsucht.

::*(JOIN) Es werden alle Benutzerkennungen in allen lokal verfügbaren Pubsets nach Dateien mit den gewünschten Eigenschaften durchsucht.

*:**:*(JOIN) Es werden alle Benutzerkennungen in allen lokal und remote zugreifbaren Pubsets nach Dateien mit den gewünschten Eigenschaften durchsucht.

()[-]*string* | [:*x*:|:*[*]:] ()[-]*string* Es werden nur diejenigen Kennungen zur Selektion herangezogen, die den String in ihrem Namen enthalten, bzw. nicht enthalten (-*string*).

Hinweise:

Normalerweise wird von CFS für jede Benutzerkennung eine Meldung ausgegeben, unter der keine Dateien der gewünschten Art gefunden wurden: "NO FILES FOUND FOR USER-ID xxxxxx". Durch Setzen des Prozeßschalters 4 (/SETSW ON=4) können diese Meldungen am Bildschirm unterdrückt werden. Um auch ohne SETSW ON=4 die NO-FILES-FOUND Meldungen zu unterdrücken, muß der Systemverwalter in der PAM-Seite 6 von CFS bzw. im Modul CFSMAIN einen Indikator setzen (Hinweise hierzu siehe Source-Listing von CFSMAIN).

Um die Suchzeiten für die Dateiselektion zu minimieren, kann die JOIN-Funktion mit der User Option NO (Names Only) gekoppelt werden.

Die gefundenen Dateien werden pro Benutzerkennung nach den im Feld SORT-OPTION angegebenen Bedingung sortiert. Durch das Kommando SORT *opt* kann die Dateienliste insgesamt nach einem vorgegebenem Kriterium sortiert werden.

USER-ID : (JOIN) darf nicht zusammen mit USER OPTION : OPEN angegeben werden.

Beispiel:

`()UTM`
Durch diese Vorgabe werden bei der Selektion alle Benutzerkennungen berücksichtigt, die in ihrem Namen irgendwo die Zeichenfolge UTM enthalten.

[:*cat-with-wildcard*:] [*user-id-with-wildcard*]

cat-with-wildcard

Cat-ID Auswahl mit Namensmuster gemäß FSTAT- (SHOW-FILE-ATTR) Syntax.
Das Konstrukt <*x*:*y*> darf nicht angegeben werden.

user-id-with-wildcard

User-ID Auswahl mit Namensmuster gemäß FSTAT- (SHOW-FILE-ATTR) Syntax.

`:C/<1,2,3>:SYS*`
Durch diese Vorgabe werden bei der Selektion alle Benutzerkennungen selektiert, die mit SYS beginnen. Außerdem werden alle Cat-ID's berücksichtigt, die mit C beginnen, an der zweiten Stelle ein beliebiges Zeichen enthalten und an der dritten Stelle eine der Ziffern 1 bis 3 enthalten.

(*datei*) Die USER-IDs (auch mit Suchmustern) werden aus der angegebenen Datei gelesen. Die Datei muß die Eigenschaften Recform=V und Fcbtype=SAM besitzen.
Die Datensätze zur Auswahl von Cat-ID's und/oder Benutzerkennungen dürfen auch Wildcards enthalten (siehe oben). Die Sätze können auch länger als 14 Stellen angegeben werden (Länge des Feldes User-ID in der Selektionsmaske).

Beispiel:

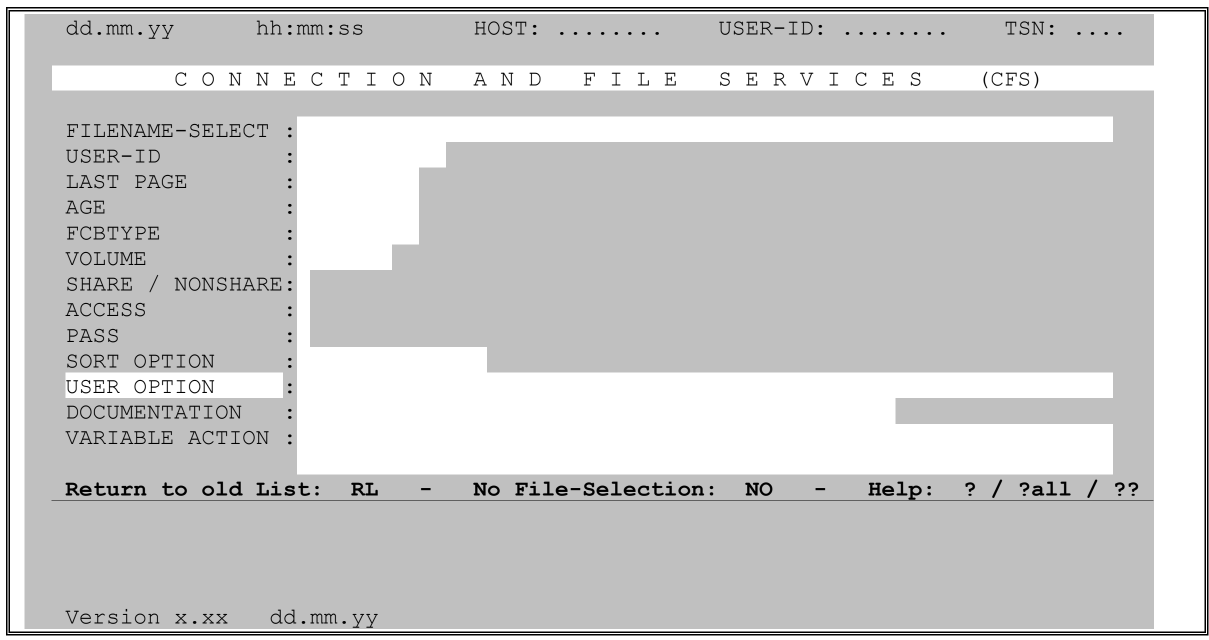
Die Datei mit dem Namen CFS.USER enthält 4 Datensätze mit folgendem Inhalt:

```
:G:TT08
:OPG:T*
CFS
:<1,2>:S43
```

`NPSAP;(CFS.USER)`
Durch dieses NP-Kommando werden alle Dateien ausgewählt, die in ihrem Namen die Zeichenfolge 'SAP' enthalten und unter :G:$TT08., :OPG:$T*., $CFS., :1:$S43. oder :2:$S43. katalogisiert sind.



Selektionsfeld : USER OPTION



```
dd.mm.yy       hh:mm:ss        HOST: ........    USER-ID: ........    TSN: ....

        C O N N E C T I O N   A N D   F I L E   S E R V I C E S    (CFS)

FILENAME-SELECT :
USER-ID         :
LAST PAGE       :
AGE             :
FCBTYPE         :
VOLUME          :
SHARE / NONSHARE:
ACCESS          :
PASS            :
SORT OPTION     :
USER OPTION     :
DOCUMENTATION   :
VARIABLE ACTION :

Return to old List:  RL   -   No File-Selection:  NO   -   Help:  ? / ?all / ??

Version x.xx   dd.mm.yy
```

Auswahl aller geöffneten Dateien im System

OPEN

Diese User Option zeigt alle Dateien im System, die zum aktuellen Zeitpunkt in einem beliebigen Modus eröffnet sind, sowie die nicht eröffneten, jedoch mit einem Linknamen verknüpften Dateien. Die Selektion erstreckt sich über die Task-File-Tables aller TSN's. Die User Option ist nur unter TSOS aufrufbar und benutzt den SVC 79. Die User Option OPEN kann in eingeschränktem Umfang auch für Nicht-TSOS Benutzer bereitgestellt werden, falls der Systemverwalter unter TSOS einen CFS-Holdertask startet (/ENTER CFS.S.LMSLIB(CFSHT)). Für weitere Informationen siehe Seite A600.
Aufgrund einer zusätzlichen Option wird neben der TSN und dem Open-Modus die Anzahl der in einem vorgegebenen Zeitintervall verbrauchten IO's für jede Datei angezeigt.

Selektionsparameter: [**S**,] [**I**|**O**|**RL**|**E**|**N**|**PR**|**PL**|**SEC**[**EX**]|**ALL**] [,**TSN=***tsn*| (*tsn1*, *tsn2*,...)] [,**USER=** *user-id*] [,**POOL=***isam-pool*] [,**LINK** [=*link*]] [,**D-IO**[**S**]|**A-IO**[**S**] [><=*x*] [,INT= *sec*] [, RW]]

- S neben Benutzertasks werden auch die von Systemtasks eröffneten Dateien berücksichtigt. Die S-Option alleine ist gleichbedeutend mit IOEN,S. Durch OPEN S werden somit alle aktiven und inaktiven TFT-Einträge angezeigt.
- I alle mit Input eröffneten Dateien
- O alle mit Output, Outin, Inout, Update, Extend eröffneten Dateien
- RL wie O, es werden jedoch nur die mit SHARUPD=YES eröffneten ISAM-Dateien ausgewählt, für die eine Satz- oder Blocksperre besteht. In der Spalte Open-Modus wird dies mit #IO angezeigt zusammen mit der entsprechenden TSN.
- E alle mit Exec eröffneten Dateien (geladene Programme, SDF-SyntaxDateien)
- N alle mit File zugewiesenen, jedoch noch nicht bzw. nicht mehr eröffneten Dateien (inaktive TFT-Einträge)
- PR alle durch ein Print-Kommando gesperrten Dateien (Open-Typ PR),
  alle durch ein TRANSFER-FILE gesperrten Dateien (Open-Typ FT),
  alle durch einen System-Lock gesperrten Dateien (open-Typ SY).

PL alle geöffneten PLAM-Bibliotheken, sowie die Namen der darin geöffneten Elemente. Zusätzlich werden auch TSN und Open-Modus (I/O/U/E) angezeigt.
Der Action-Code D (Display) auf geöffnete PLAM-Elemente ist nicht möglich.
SEC alle mit einem Secure-Kommando shareable oder exclusiv reservierten Dateien
SECEX alle mit einem SECURE-Kommando exclusiv reservierten Dateien
SECEX ist eine Teilmenge von SEC (s.o.)
ALL Abkürzung für IOPRPLSEC, falls der Parameter TSN= nicht angegeben wurde.
Abkürzung für IOENPRPLSEC, falls der Parameter TSN= angegeben wurde.

Standard:

IOPRSEC, falls der Parameter TSN= nicht angegeben wurde.
IOENPRSEC, falls der Parameter TSN= angegeben wurde.

Die Selektionsbedingungen I O E N PR PL SEC[EX] ALL dürfen in beliebigen Kombinationen angegeben werden (z.B. IO, OIE, IOENSEC, ALLE).

TSN=*tsn*|(*tsn1*,*tsn2*,...)

Anzeige aller TFT-Einträge zu der angegebenen TSN bzw. zu der Liste der TSNs. Für *tsn* kann auch die Bezeichnung einer Systemtask angegeben werden. Beispiel: TSN=CLOG.

USER=*user-id*

Die Angabe einer Benutzerkennung (z.B. TSOS) bewirkt, daß nur diejenigen Tasks für die Auswahl berücksichtigt werden, die unter der angegebenen User-Id angeloggt sind.

POOL=*isam-pool*

Durch Angabe eines Pool-Namens werden alle NK-ISAM Dateien ausgewählt, die an diesem Pool angeschlossen sind.

LINK Es werden die zum Eröffnen der Dateien verwendeten Linknamen angezeigt. Standard: Es werden keine Linknamen ausgegeben.

link Es werden nur die mit dem angegebenen Linknamen eröffneten Dateien selektiert. Beispiel: LINK=ECERDLOD selektiert alle geladenen Programme.

D-IO [><=*x*]

Delta IO. In der AGE-Spalte der Dateienliste wird die <u>Anzahl der IO's</u> angezeigt, die in einem bestimmten Zeitintervall (Standard=30 Sekunden) für jede einzelne Datei angefallen sind. Im Unterschied zu der Option D-IOS (siehe unten) werden Dateien, die von mehreren Tasks eröffnet sind, in der Liste mehrfach aufgeführt zusammen mit der entsprechenden TSN und der Anzahl der IO's. Durch ><=*x* werden alle Dateien ausgewählt, die mehr als, weniger als oder genau *x* IO's verbraucht haben. Beispiel: `OPENS D-IO>0`

D-IOS [><=*x*]

Delta IO Summe. In der AGE-Spalte der Dateienliste wird die <u>Anzahl der IO's</u> angezeigt, die in einem bestimmten Zeitintervall (Standard=30 Sekunden) für jede einzelne Datei angefallen sind. Für Dateien, die von mehreren Tasks eröffnet sind, wird der Summenwert der IO's über alle Tasks ausgegeben. In der Spalte TSN wird dann '****' eingetragen, da sich die Anzahl der IO's nicht auf eine Task bezieht. Durch ><=*x* werden alle Dateien ausgewählt, die mehr als, weniger als oder genau *x* IO's verbraucht haben.

Beispiel:

User-Option: `OPENS,D-IOS>0` und Sort-Option: `AGE,D`
Die eröffneten Dateien werden nach der Anzahl der verbrauchten IO's absteigend sortiert.

INT=*sec*

Länge des Zeitintervalls für D-IO in Sekunden. Standard: INT=30.
Um die IO-Werte ausgeben zu können, wird CFS für die angegebene Anzahl von Sekunden in einen Wartezustand versetzt.

RW In den beiden ersten Anzeigespalten der Dateienliste wird für jede Datei anstelle von LASTP und TYP die Anzahl der verbrauchten Read-/Write-IO's angezeigt.

Ohne Angabe von Selektionsbedingungen werden alle im Input- oder Output-Modus eröffneten Dateien angezeigt. Nicht ausgegeben werden Dateien, die

- von Systemtasks eröffnet sind
- im EXEC-Modus eröffnet sind (geladene Programme, SDF-Syntaxdateien)
- Dateien, die mit einem File-Kommando zugewiesen, jedoch nicht eröffnet sind

A-IO [><=*x*]

Accumulated IO. In der AGE-Spalte der Dateienliste wird die Anzahl der IO's angezeigt, die seit dem Eröffnen der Datei durch die entsprechende TSN angefallen sind. Im Unterschied zu der Option A-IOS (siehe unten) werden Dateien, die von mehreren Tasks eröffnet sind, in der Liste mehrfach aufgeführt. Durch ><=*x* werden alle Dateien ausgewählt, die mehr als, weniger als oder genau *x* IO's verbraucht haben. Beispiel: `OPENS A-IO>100`
Die Option A-IO und A-IOS (siehe unten) steht erst ab OSD V3.0 zur Verfügung.

A-IOS [><=*x*]

Accumulated IO Summe. In der AGE-Spalte der Dateienliste wird die Anzahl der IO's angezeigt, die seit dem Eröffnen der Datei durch die entsprechende TSN angefallen sind. Für Dateien, die von mehreren Tasks eröffnet sind, wird der Summenwert der IO's über alle Tasks ausgegeben. In der Spalte TSN wird dann '****' eingetragen, da sich die Anzahl der IO's nicht auf eine Task bezieht. Durch ><=*x* werden alle Dateien ausgewählt, die mehr als, weniger als oder genau *x* IO's verbraucht haben.

Ausgaben der User Option OPEN:

In der Spalte User Option der Dateienliste wird die TSN des Prozeßes angezeigt, der die Datei eröffnet bzw. der ein File-Kommando für die Datei abgesetzt hat.
Die Angabe *REM in der TSN-Spalte besagt, daß die Datei von einem anderen Rechner eröffnet wurde. Der eigene Host muß hierbei die Rolle des Masters für den Shared Pubset erfüllen.
Neben der TSN wird der Modus angezeigt, in dem die Datei eröffnet ist:

I	Input
O	Output
IO	Inout
OI	Outin
U	Update
E	Extend
EX	Exec
R	Reverse
⊔	Inaktiver TFT-Eintrag. Datei ist gegenwärtig nicht eröffnet
S	Datei ist mit einem Secure-Kommando shareable reserviert

SE Datei ist mit einem Secure-Kommando exclusiv reserviert
PR Datei wird mit PRINT ...,LOCK=YES ausgedruckt

Ein Stern vor dem Open-Modus besagt, daß die Datei mit SHARUPD=YES eröffnet ist.

Das Zeichen (#) vor dem Open-Modus IO besagt: Die Datei ist mit SHARUPD=YES eröffnet und von der links stehenden TSN wird eine Satz-/Blocksperre gehalten.

Bei geöffneten NK-ISAM Dateien wird in der ersten Spalte der Dateienliste (LASTP) der Name des verwendeten ISAM-Pools angezeigt.

Hinweise:

Falls keine SORT OPTION angegeben wurde, werden die durch die User Option OPEN selektierten Dateien in der Reihenfolge der erzeugenden TSNs und hierbei in der Reihenfolge der abgesetzten FILE-Kommandos aufgelistet. Eine Datei kann mehrmals in der Liste vorkommen, falls sie von verschiedenen Tasks eröffnet wird.

Die Auswahl von Dateien über die User Option OPEN kann mit allen anderen Bedingungen, insbesondere mit der Auswahl über FILENAME-SELECT kombiniert werden. Sollen z.B. alle TSNs gesucht werden, unter denen eine bestimmte Datei eröffnet wurde, so ist im Feld FILENAME-SELECT der vollqualifizierte Dateiname und der Parameter ,Q anzugeben.

Nach der Selektion von Dateien mit der User Option OPEN kann in der Dateienliste mit dem Action-Code S ein Status Kommando der ausgewiesenen TSN angefordert werden.

USER OPTION : OPEN darf nicht zusammen mit USER-ID : (JOIN) angegeben werden.

IO-Auswertung mit der Option D-IO: Während der Dauer der Auswertung kann das Programm KDCMON nicht verwendet werden. Umgekehrt kann während einer KDCMON-Messung die Option D-IO nicht verwendet werden. Eine D-IO Auswertung kann auch nicht von mehreren Benutzern gleichzeitig aufgerufen werden. In diesem Fall wird die Meldung "KDCMON active" ausgegeben. Sollte diese Meldung ausgegeben werden, wobei das Programm KDCMON aber nachweislich nicht geladen ist, so kann die Lock-Situation beseitigt werden, indem anstelle von D-IO, die Option X-IO verwendet wird.

Eine mit D-IO erzeugte Dateienliste kann mit dem CFS-Kommando SORT D-IO,D oder SORT AGE,D nach der Anzahl der verbrauchten IO's absteigend sortiert werden. Die gleiche Sortierung kann auch über das Feld SORT OPTION der Selektionsmaske erreicht werden. Hier ist als Sortierkriterium AGE,D anzugeben.

Zur kontinuierlichen Verfolgung des IO-Verhaltens einer bestimmten Datei oder einer Task kann die Prozedur FILEIO in der CFS.USERLIB.BEISPIEL aufgerufen werden (DO $*user-id*.CFS.USERLIB.BEISPIEL(FILEIO),?).

Geöffnete SYSOUT-/SYSLST-Dateien, die in einem Element einer PLAM-Bibliothek abgelegt werden, konnten bisher von der User Option OPEN nicht erfaßt werden.
Ab SYSFILE V13.0A (OSD 4.0) steht das Kommando **MOD**IFY**-SYSF**ILE**-OPT**IONS **LIB**RARY**-OUT**PUT**-COPY**=**Y**ES zur Verfügung. Damit wird der Inhalt eines Sysfile PLAM-Elements zusätzlich in eine temporäre Datei auf Platte geschrieben. Zum Zeitpunkt, wo das PLAM-Element geschlossen wird, wird auch die temporäre Sysfile-Datei gelöscht. Die temporäre Sysfile-Datei wird von der User Option OPEN angezeigt. Der Inhalt der Datei kann mit dem Action-Code D sichtbar gemacht werden.

Die User Option OPEN kann auch von Normalbenutzern unter Kennungen ungleich TSOS angewendet werden. Die Anzeige erstreckt sich hierbei jedoch nur die offenen Dateien der eigenen Benutzerkennung. Der Inhalt schreibend geöffneter SAM-/PAM-Dateien kann mit dem Action-Code D (Display) sichtbar gemacht werden. Damit diese Möglichkeiten genutzt werden können, muß der Systemverwalter unter TSOS einen CFS-Holdertask starten: /ENTER $CFS.CFS.S.LMSLIB(CFSHT). Näheres hierzu in der Inline-Beschreibung der Enter-Datei.

Beispiele:

`OPEN TSN=9011`
Alle zum Zeitpunkt der Selektion existierenden TFT-Einträge der TSN 9011.

`OPEN TSN=9011,D-IO>0,INT=60`
Alle Dateien der TSN 9011, die in den letzten 60 Sekunden IO's verbraucht haben. In der Spalte D-IO der Dateienliste wird die Anzahl der IO's angezeigt. Vor Ausgabe der Dateienliste wird CFS für 60 Sekunden in einen Wartezustand versetzt.

`OPEN O`
Alle zum Zeitpunkt der Selektion eröffneten Ausgabedateien.

`OPEN PR`
Alle durch ein Print-Kommando mit Parameter LOCK=YES gesperrten Dateien.

`OPEN SEC`
Alle mit einem Secure-Kommando angeforderten oder exklusiv belegten Dateien.

`OPEN TSN=M,LINK`
Alle von der Systemtask mit TSN M eröffneten Dateien zusammen mit den zugehörigen Linknamen.

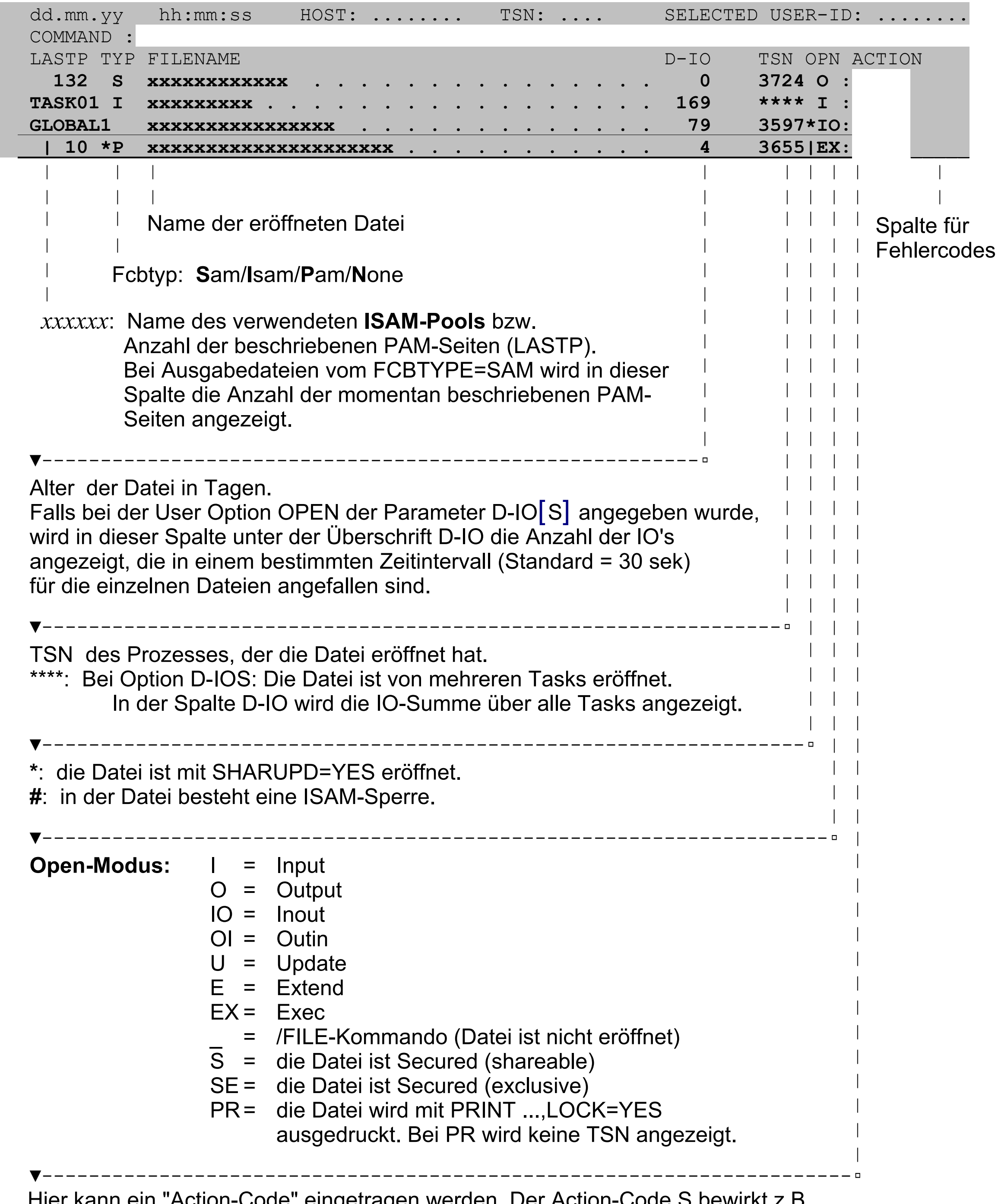
`OPENS,TSN=CLOG`
Es wird die aktive Conslog-Datei ausgegeben. Um den Inhalt der Datei anzuschauen, ist In der Action-Spalte der Action-Code 'D' einzutragen.

`OPEN IOEN,LINK=CFSCMD`
Alle aktiven bzw. nicht aktiven TFT-Einträge mit dem Linknamen CFSCMD.

`OPEN USER=S43`
Alle eröffneten Dateien von Tasks, die unter S43 angeloggt sind.

`OPEN POOL=$TASK01`
Alle eröffneten NK-ISAM Dateien, die an dem Pool $TASK01 angeschlossen sind.

Format der Dateienliste für User Option OPEN:



Hier kann ein "Action-Code" eingetragen werden. Der Action-Code S bewirkt z.B. die Ausgabe der Status-Informationen zur angezeigten TSN. Der Action-Code D (Display) kann auch auf SAM-Dateien angewendet werden, die im Output-Modus eröffnet sind, z.B. Ablaufprotokolle, CONSLOG-Datei.

Auswahl aller aktiven Monitor Jobvariablen

ONEVT

Es werden alle Monitor Jobvariablen im System angezeigt, die zum aktuellen Zeitpunkt mit einem ONEVT-Kommando bzw. Makro überwacht werden. Die User Option ist nur unter TSOS aufrufbar und benutzt den SVC 79. Die User Option ONEVT kann in eingeschränktem Umfang auch für Nicht-TSOS Benutzer bereitgestellt werden, falls der Systemverwalter unter TSOS einen CFS-Holdertask startet (/ENTER CFS.S.LMSLIB(CFSHT)).

Selektionsparameter: [**S**,] [,**TSN=***tsn*] [,**USER=***userid*]

S neben Benutzertasks werden auch die von Systemtasks angesprochenen Monitor Jobvariablen berücksichtigt.

TSN=*tsn*
Anzeige aller aktiven Monitor Jobvariablen zu der angegebenen TSN.

USER=*userid*
Die Angabe einer Benutzerkennung (z.B. TSOS) bewirkt, daß nur diejenigen Tasks für die Auswahl der Monitor Jobvariablen berücksichtigt werden, die unter der angegebenen User-Id angeloggt sind.

Ausgaben der User Option ONEVT:

In der Spalte User Option der Dateienliste werden die TSN's der Prozesse angezeigt, die die Jobvariable mit einem ONEVT-Kommando überwachen. Falls mehrere Tasks eine Jobvariable überwachen, wird der JV-Name entsprechend oft in der Dateienliste aufgeführt.

Hinweise:

Falls keine SORT OPTION angegeben wurde, werden die durch die User Option ONEVT selektierten Jobvariablen in der Reihenfolge der erzeugenden TSNs aufgelistet. Eine Jobvariable kann mehrmals in der Liste vorkommen, falls sie von verschiedenen Tasks überwacht wird.

Die Auswahl von Jobvariablen über die User Option ONEVT kann mit allen anderen Bedingungen, insbesondere mit der Auswahl über FILENAME-SELECT kombiniert werden.

Nach der Selektion von Jobvariablen mit der User Option ONEVT kann in der Dateienliste mit dem Action-Code S ein Status Kommando der ausgewiesenen TSN angefordert werden.

Mit dem Action-Code OPN kann sich der Benutzer über alle Tasks (TSN's) informieren, die eine bestimmte Jobvariable überwachen.

USER OPTION : ONEVT darf nicht zusammen mit USER-ID : (JOIN) angegeben werden.

Die User Option ONEVT kann auch von Normalbenutzern unter Kennungen ungleich TSOS angewendet werden. Die Anzeige erstreckt sich hierbei jedoch nur die aktiven Monitor Jobvariablen der eigenen Benutzerkennung. Damit diese Möglichkeit genutzt werden kann, muß der Systemverwalter unter TSOS einen CFS-Holdertask starten: /ENTER $CFS.CFS.S.LMSLIB(CFSHT). Näheres hierzu in der Inline-Beschreibung der Enter-Datei.

Beispiele:

```
ONEVT
```
Alle per ONEVT-Kommando angesprochenen Jobvariablen über alle Benutzerkennungen.

```
ONEVT TSN=9011
```
Alle Monitor Jobvariablen, die von der TSN 9011 überwacht werden.

Auswahl aller Isam-Dateien mit unbenutzten Datenblöcken

FREE[%][ISAM][,PLAM]

Anzahl der freien PAM-Seiten einer Datei, d.h. zugewiesener, jedoch nicht belegter Bereich. Free = Allocated Space - Last Page

% Prozentanteil der freien PAM-Seiten bezüglich der Gesamtgröße der Datei.

ISAM Handelt es sich bei einer Datei um eine ISAM-Datei, so wird unter Zuhilfenahme des Kommandos SHOW-FREE-ISAM-SPACE die Anzahl der unbenutzten Datenblöcke ausgegeben. Die Option ISAM kann nur unter der Benutzerkennung TSOS angegeben werden. Das FSC-Softwareprodukt SPACEOPT V2.0 muß außerdem installiert sein.

PLAM Handelt es sich bei einer Datei um eine PLAM-Bibliothek, so wird der innerhalb der Bibliothek nicht benutzte Platz zu der Anzahl der freien Seiten hinzugezählt. Die Option PLAM kann ab LMS Version 3.0 benutzt werden.

Selektionsparameter: >*x* | <*x* | =*x*

Auswahl aller Dateien/Jobvariablen mit einem bestimmten Passwort

PSW Diese User Option zeigt nach Wahl entweder ein bestehendes WRITE-, READ- oder EXEC-Passwort für die selektierten Dateien bzw. selektiert alle Dateien, die ein angegebenes WRITE-, READ- oder EXEC-Passwort besitzen.

Pflichtparameter: **W** | **R** | **E**
Es werden die WRITE-, READ- oder EXEC-Passworte angezeigt.

Beispiel:

```
PSW W
```
Sofern vorhanden, werden die Write-Passworte der Dateien in der Liste angezeigt.

Wahlweise Selektionsparameter: [, X'*passw*'] | [, C'*passw*']
Es werden alle Dateien ausgewählt, die das angegebene Passwort besitzen.

Beispiel:

```
PSW W,X'AFFE'
```
Es werden alle Dateien ausgewählt, die das Write-Passwort X'AFFE' besitzen.

Hinweis:

Die User Option PSW ist nur in BS2000-Systemen ohne Passwortverschlüsselung anwendbar.

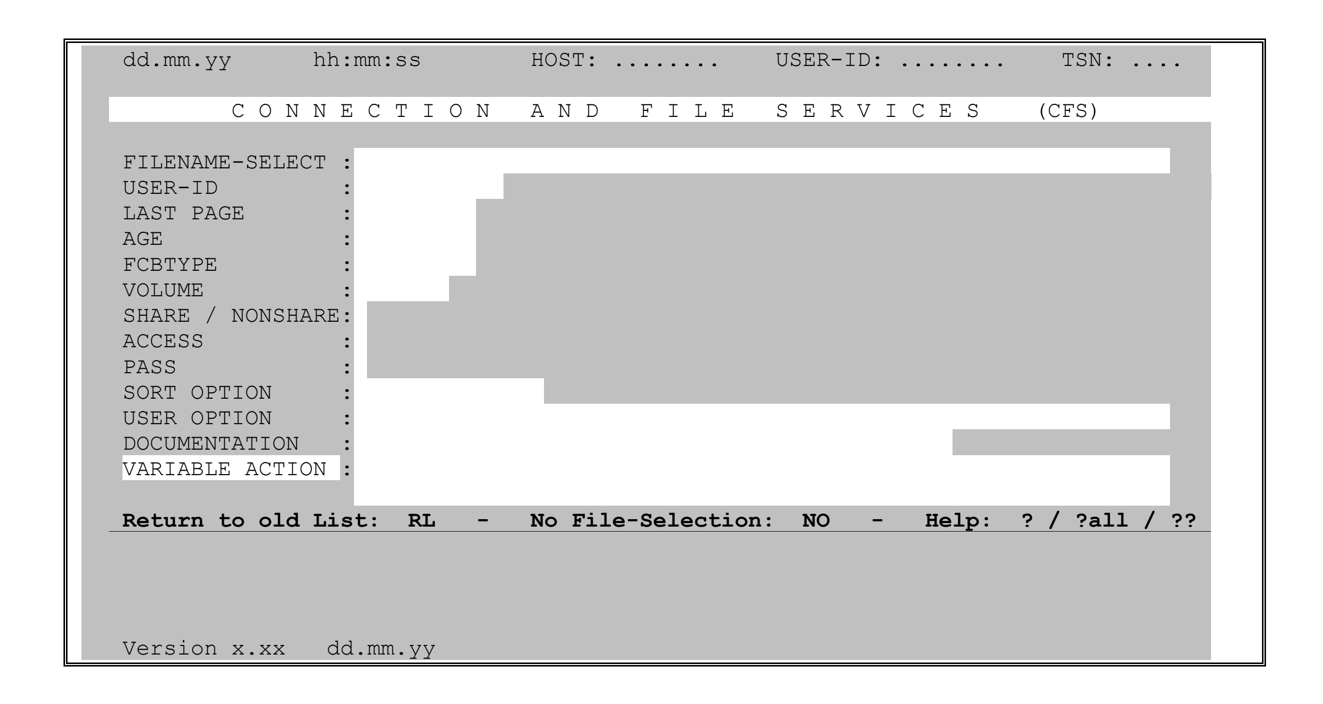
Auswahl von Dateien auf einem Volume belegter Speicherplatz

VOLSP

Es wird die Anzahl der PAM-Seiten ausgegeben, die die jeweilige Datei auf einer vorgegebenen Platte (VSN) belegt hat.

Selektionsparameter (obligatorisch): *vsn* [>*x* | <*x* | =*x*]
Mit dem optionalen Zusatz >|<|=*x* werden alle Dateien ausgewählt, die auf dem angegebenen Datenträger mehr als, weniger als oder genau *x* Seiten belegt haben.

Variable Actions



Dateien kopieren zwischen verschiedenen Benutzerkennungen/Public-Volume Sets

ONX**COPY** '*:x:*$*user-id1.*'='*:y:*$*user-id2.*' [, STD] [, KD]

Diese Variante der Variablen Action COPY hat zur Voraussetzung, daß die Dateien durch Eingabe von FSTAT $*user-id1.* im Feld FILENAME-SELECT selektiert wurden oder daß die CFN-Option (Complete FileNames) aktiv ist.

Die Variable Action bewirkt, daß alle durch X markierten Dateien aus der Kennung $*user-id1.* und PVS *:x:* in die Kennung $*user-id2.* und den PVS *:y:* kopiert werden.

ONX**COPY** '' = '$*user-id.*' [, *vsn*/*device* | *:x:* [,*smspar*]] [, STD] [, KD]

Die angekreuzten Dateien werden von der selektierten Benutzerkennung in die in der ONXCOPY-Anweisung angegebene Kennung ($*user-id.*) kopiert. Die kopierten Dateien erhalten dabei bis auf die neue Benutzerkennung die gleichen Namen wie die Ursprungsdateien.

vsn/*device* Die kopierten Dateien werden auf der angegebenen Privatplatte eingerichtet.

:x: Die kopierten Dateien werden in dem angegebenen Public-Volume-Set eingerichtet.

smspar Falls die kopierte Datei auf einem SM-Pubset (System Managed Pubset) angelegt wird, so kann mit diesem Parameter in Abhängigkeit von den Vorgaben des Systemverwalters der Ablageort (Volume-Set) der Datei genauer bestimmt werden.
Falls vor *smspar* der Parameter *:x:* nicht angegeben wurde, so bezieht sich *smspar* auf den Default-Pubset der Benutzerkennung.

STORAGE-CL=*class*
Name einer vom Systemverwalter eingerichteten Storage Klasse. Über die Storage Klasse wird das Volume-Set der Datei und damit die Speichereigenschaften wie z.B. Performance Attribute, Availability festgelegt.

VOLUME-SET=*vset*
Bestimmt das Volume-Set, auf dem die Datei innerhalb des SM-Pubsets eingerichtet werden soll. Der Systemverwalter kann mit *CONTROL bzw. mit der Angabe einer ein- bis vierstelligen Cat-Id *vset* (ohne ':') einen Volume-Set explizit angeben.
Der nichtprivilegierte Benutzer kann einen Volume-Set nur dann angeben, wenn ihm der Systemverwalter das Recht zur physikalischen Allokierung gegeben hat.

WORK-FILE=YES|NO
Gibt an, ob die Datei eine Arbeitsdatei ist, die von der Systemverwaltung zu einem bestimmten Zeitpunkt gelöscht werden darf. Work-Dateien werden auf einem Volume-Set abgelegt, für das der Systemverwalter die Nutzungsart WORK vorgegeben hat.

STD

Vorbemerkung: Beim Kopieren von Dateien mit der Variablen Action ONXCOPY werden die Schutzattribute der Quelldateien wie z.B. ACCESS=READ, RETPD oder Passworte auf die Zieldateien übertragen (COPY ..,SAME). Unter TSOS setzt CFS zusätzlich alle für den lesenden Zugriff auf die Quelldateien und den schreibenden Zugriff auf die Zieldateien notwendigen Passworte ab. Weitere Schutzattribute der mit Copy zu überschreibenden Zieldateien wie ACCESS=READ und RETPD>0 werden von CFS zum Zwecke des Kopierens kurzzeitig zurückgesetzt. Gleiches gilt auch für die Variable Action ONXMOVE.

Der Operand STD in der Variablen Action ONXCOPY bewirkt zweierlei:

1) Die Quelldateien werden nicht mit dem Parameter "COPY ...,SAME" kopiert, d.h. die Schutzattribute der Quelldateien werden nicht auf die Zieldateien übertragen.

2) Die Schutzattribute der Dateien werden durch CFS auch unter der Kennung TSOS nicht umgangen. Es können somit nur die Dateien kopiert werden, für die der Benutzer zuvor die zum Lesen und Überschreiben benötigten Passworte abgesetzt hat. Darüber hinaus können Zieldateien durch COPY nicht überschrieben werden, falls diese bereits existieren und mit ACCESS=READ oder RETPD>0 geschützt sind.

KD

Keep Date. Beim Kopieren wird das Change Date der Originaldateien auf die Zieldateien übertragen. Die kopierten Dateien erhalten somit das gleiche Change Date wie die Ursprungsdateien und nicht das Datum des heutigen Tages. Die KD-Option ist unwirksam, falls Dateien auf eine Privatplatte kopiert werden.
Bei eingeschaltetem KLA-Modus (Kommando (KLA)) wird zusätzlich das Last Access Date der Eingabe- und Ausgabe-Datei wieder hergestellt.

Beispiele: $ABC. = selektierte Benutzerkennung.

`ONXCOPY''='$USER2.'`	$ABC.*name*	-->	$USER2.*name*
`ONXCOPY':X:'=':Y:'`	:X:$ABC.*name*	-->	:Y:$ABC.*name*
`ONXCOPY''='A.'`	$ABC.*name*	-->	$ABC.A.*name*
`ONXCOPY''=''`	$ABC.*name*	-->	*name* (eigene Kennung)
`ONXCOPY' '='.X'`	$ABC.*name*	-->	$ABC.*name*.X
`ONXCOPY' '='.X',:A100:`	[:*xxx*:]$ABC.*name*	-->	:A100:$ABC.*name*.X

Hinweise:

Mit der Variablen Action ONXCOPY können unter TSOS auch Katalogeinträge für Banddateien zwischen verschiedenen Kennungen und/oder Public-Volume-Sets kopiert werden, ohne daß die entsprechenden Bandvolumes zum Mounten angefordert werden.

Das automatische Absetzen von Passworten ist auch in BS2000-Systemen mit Passwortverschlüsselung anwendbar (ENCRYPT=YES).

ONX**COPY** *:x:* [, STD] [, KD]

Die angekreuzten Dateien werden auf den angegebenen Pubset kopiert.
Die Dateien werden auf dem neuen Pubset unter der gleichen Kennung wie die Ursprungsdateien angelegt.

Diese Variante der Variablen Action ONXCOPY hat die gleiche Wirkung wie ONXMOVE, mit dem Unterschied, daß die Ursprungsdateien nicht gelöscht werden.

:x: Public-Volume-Set der Zieldateien.

Beispiel: ausgewählte Dateien: NP USER-ID=:X:USER2

`ONXCOPY :RST:` :X:$USER2.*name* --> :RST:$USER2.*name*

ONX**COPY** '$*user-id*.' | '*:x:*'

Die angekreuzten Dateien werden überschrieben durch gleichnamige Dateien aus der Benutzerkennung $*user-id*. bzw. aus dem PVS *:x:*.

$*user-id*. Benutzerkennung der Quelldateien.

:x: Public-Volume-Set der Quelldateien.

Beispiele: ausgewählte Dateien: NP USER-ID=:R:
d.h. alle Dateien der eigenen Kennung auf PVS R.

`ONXCOPY':X:'` :X:*name* --> :R:*name*

`ONXCOPY'$USER2.'` $USER2.*name* --> :R:*name*

Dateien umbenennen von einer User-Id in eine andere User-Id

ONX**REN** "='$*user-id2*.'
ONX**REN** '$*user-id1*.'='$*user-id2*.'

Die Variable Action bewirkt, daß alle durch X markierten Dateien von ihrer ursprünglichen Kennung in die neue Kennung $*user-id2*. umbenannt werden. Da hierbei keine Daten bewegt werden, kann die Umbenennung der Dateien nur innerhalb des gleichen PVS erfolgen.

Falls die Namen der Dateien in der Dateienliste ohne die Benutzerkennung angezeigt werden, so ist gemäß der allgemeinen Namenstransformationsregeln in CFS anzugeben: ONX**REN** "='$*user-id2*.'

Falls die Namen der Dateien in der Dateienliste die Benutzerkennung enthalten ($*user-id1.*), so ist anzugeben: ONX**REN** '$*user-id1.*'='$*user-id2.*'

Dateien auf Privatplatten und Dateigenerationen können mit der Variablen Action ONXREN nicht in eine andere Benutzerkennung umbenannt werden.

Katalogeinträge in Datei schreiben

ONX**LISTCAT** *ausdat* [, VERSION=0 | 2 | **3**] [, E]

Diese Variable Action schreibt für die mit X gekennzeichneten Dateien/Jobvariablen den vollständigen Katalogeintrag als einen Satz variabler Länge in die Datei *ausdat*. Eventuelle vorhandene Passworte werden im Katalogauszug beibehalten und **nicht** auf binär Null gelöscht.

VERSION=3 Dieser Parameter steht nur dem Systemverwalter unter der Kennung TSOS zur Verfügung. Es wird der physikalische Katalogeintrag in die Ausgabedatei geschrieben. Das Layout wird durch die DSECT DCACATE MF=D,CETYPE=*FILE | *JV beschrieben.
Eine mit VERSION=3 erstellte Listcat-Datei kann nicht mit dem Kommando RECAT verarbeitet werden.

Das Kommando RECAT (siehe Seite A595) kann die mit LISTCAT erzeugte Katalogauszugsdatei verarbeiten. RECAT vergleicht die in der LISTCAT-Datei festgehaltenen Dateiattribute inclusive Passworten mit den aktuellen Dateiattributen. Bei Ungleichheit werden geeignete CAT-Kommandos abgesetzt, um die früheren Dateiattribute wieder herzustellen.

Bibliotheken / Dateien reorganisieren

ONX**REORG** ['*str1*'='*str2*'] [,*:x:* | *vsn*/*device* | PAD=*nn* | NR | DF[O] | SPACE=(*xxxx*,*yyy*) | **KD**]

Die angekreuzten Bibliotheken/Dateien werden reorganisiert, indem alle Elemente/ Datensätze in neue Bibliotheken/Dateien übertragen werden. Für eine ausführliche Beschreibung aller Parameter der Variablen Action REORG wird auf Seite **Fehler! Textmarke nicht definiert.** verwiesen.

KD **Keep Date**. Die reorganisierten Dateien/Bibliotheken erhalten das Change Date der Ursprungsdateien und nicht das Datum des heutigen Tages, wie dies ohne Angabe der KD-Option der Fall wäre. Die KD-Option ist unwirksam, falls Dateien/ Bibliotheken auf eine Privatplatte reorganisiert werden.
Die KD-Option bewirkt außerdem, daß die Backup Version # und die Save Version # der Datei beibehalten werden. Dies ist für den Archive-Sicherungsstatus von Bedeutung.

Dateien auf anderen Pubset moven und Change Date übernehmen

ONX**MOVE** [*:x:*] [,*smspar*] | DEFCAT [, KD]

:x: | DEFCAT Die vom Benutzer mit dem Action-Code X bezeichneten Dateien werden unter gleichem Namen auf den angegebenen Public-Volume-Set bzw. auf den Default-Pubset der jeweiligen Benutzerkennung übertragen.

KD **Keep Date**. Beim Übertragen auf das neue Pubset wird der ursprüngliche Wert des Change Date der Dateien beibehalten. Die KD-Option ist unwirksam, falls Dateien auf eine Privatplatte übertragen werden.
Bei eingeschaltetem KLA-Modus (Kommando (KLA)) wird zusätzlich das Last Access Date wieder hergestellt.

Hinweis:

Mit der Variablen Action ONXMOVE können unter TSOS auch Katalogeinträge für Banddateien zwischen verschiedenen Public-Volume-Sets übertragen werden, ohne daß die entsprechenden Bandvolumes zum Mounten angefordert werden.

Passworte für passwortgeschützte Dateien/Bibl.Elemente/Jobvariablen setzen

ONX**SETPSW** Diese Variable Action setzt für die mit X gekennzeichneten Dateien / PLAM-Bibliothekselemente / Jobvariablen die zum uneingeschränkten Zugriff notwendigen PASSWORD-Kommandos.

Die Variable Action ist auch in BS2000-Systemen mit Passwortverschlüsselung anwendbar (ENCRYPT=YES). Das Absetzen der Passworte für geschützte PLAM-Bibliothekselemente ist ab LMS Version 3.0 möglich.

ONX**SETPSW** [REL=Y]

Für die mit X gekennzeichneten Dateien / PLAM-Bibliothekselemente / Jobvariablen werden die mit ONXSETPSW bekannt gemachten Passworte wieder aus der Passwort Tabelle entfernt (PASSWORD ...,REL=Y).

Dateien entsperren / ISAM-Dateien reparieren

ONX**VERIFY** [REP[AIR]=NO|YES|ABS|CHECK]

Die mit X gekennzeichneten Dateien werden entsperrt, d.h. es werden evtl. bestehende Locks zurückgesetzt. ISAM-Dateien, die aufgrund eines Systemabsturzes nicht geschlossen wurden, können in einen konsistenten Zustand gebracht werden (REPAIR=YES). Die Werte des Parameters REPAIR haben die gleiche Bedeutung wie in dem BS2000-Kommando VERIFY.

Die Variable Action ONXVERIFY bietet die gleiche Funktionalität wie die BS2000-Kommandos VERIFY bzw. REPAIR-DISK-FILES. Im Unterschied zu diesen wird beim Entsperren von Dateien keine Rückfrage an den Benutzer gestellt. Die Variable Action ist außerdem auch im Batch anwendbar.

Action-Codes

```
  dd.mm.yy   hh:mm:ss    HOST: ........   TSN: ....     SELECTED USER-ID: ........
  COMMAND :
  LASTP TYP FILENAME                                       AGE PASS LACC  ACTION
     30 *I  xxxxxxxxxxxx  . . . . . . . . . . . . . . .  143 *      62 :
     77 *P  xxxxxxxxxxxx  . . . . . . . . . . . . . . .  143 *      62 :
      9 *P  xxxxxxxxxx  . . . . . . . . . . . . . . . .  143 *      62 :
     49 *P  xxx . . . . . . . . . . . . . . . . . . . .    4 . . .   2 :
     61 *I  xxxxxxxxxxxxxxxx  . . . . . . . . . . . . .    4 . . .   2 :
     10 *S  xxxxxxxxxxxxxxxx  . . . . . . . . . . . . .    4 . . .   4 :
     73  S  xxxxxxxxxxxxxxxxxxx . . . . . . . . . . . .    4 . . .   2 :
     29  S  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .    4 . . .   2 :
      0  N  xxxxxxxxxxxxxxxxxxxxx . . . . . . . . . . .      . . .   0 :
    245 *P  xxxxxxxxxxx . . . . . . . . . . . . . . . .    4 . . .   2 :
     49  P  xxxxxx  . . . . . . . . . . . . . . . . . .    2 . . .   0 :
    237 *P  xxxxxxxxxxxxx . . . . . . . . . . . . . . .    4 . . .   4 :
     12  S  xxxxxxxxxxxxxxx . . . . . . . . . . . . . .    8 . . .   7 :
      1  S  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .   45 . . .   4 :
      1  S  xxxxxxxxxxxxxxxxxxxx  . . .  . . . . . . . .   42 . . .   4 :
     14 *P  xxxxx . . . . . . . . . . . . . . . . . . .  290   RW   62 :
      9 *P  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .  148    W   53 :
      9  P  xxxxxxxxxxxxxxxxx . . . . . . . . . . . . .  148 *       1 :
  *TAPE  S  xxxxxxxxxxx . . . . . . . . . . . . . . . .    4 . . .   4 :
     11 *P  xxxxxxxxxxxxxxxxxxxxxxx . . . . . . . . . .  148 *ERW   62 :
  List continues  (P=1/T=187/H=0)           For Help:  ? / ?cmd / ?all / ??
```

Kopieren von Dateien mit Beibehaltung des Change Date

CKD

Copy + Keep Date. Beim Kopieren wird das Change Date der Originaldatei auf die Zieldatei übertragen. Ansonsten wirkt der Action-Code CKD wie CS (Copy Same). Ab OSD V8.0 kann der Action-Code CKD auch unter Benutzerkennungen ungleich TSOS angegeben werden.

Inhalt von im Output-Modus eröffneten SAM-/PAM-Dateien anzeigen

D

Falls mit der User Option OPEN [S] alle offenen Dateien ausgewählt wurden (siehe Seite A574), so kann mit dem Action-Code D der aktuelle Inhalt dieser Dateien angezeigt werden. Das gilt insbesondere für SAM oder PAM-Dateien die im Output-Modus eröffnet sind (z.B. SYSOUT-/SYSLST-Dateien, CONSLOG-Datei).

Hinweise:

Der Action-Code D bewirkt das Übertragen der momentanen Inhalte der offenen Datei in eine temporäre Hilfsdatei, die von CFS angezeigt wird. Durch das Kommando D ist es auf einfache Weise möglich, den Inhalt der Hilfsdatei auf den neuesten Stand zu bringen. Die Kommandofolge D;++ bewirkt, daß das aktuelle Ende der SAM-Datei angezeigt wird.

Standardmäßig kann der Inhalt von schreibend eröffneten ISAM-Dateien nicht, bzw. nur bei SHARUPD=YES angezeigt werden. Mit Hilfe einer Sonderoption ist es jedoch möglich, den Inhalt dieser ISAM-Dateien im PAM-Format anzuzeigen. Hierzu ist vor Absetzen des Action-Codes D einmalig das folgende CFS-Kommando einzugeben: M L'29AD','X'

Bei im Output-Modus eröffneten SAM-/PAM-Dateien zeigt die Angabe in der Spalte LASTP der Dateienliste die Anzahl der zum gegenwärtigen Zeitpunkt beschriebenen PAM-Seiten an.

Der Action-Code D auf offene SAM-/PAM-Dateien kann auch von Normalbenutzern unter Kennungen ungleich TSOS angewendet werden. Damit diese Möglichkeit genutzt werden kann, muß der Systemverwalter unter TSOS einen CFS-Holdertask starten: /ENTER $CFS.CFS.S.LMSLIB(CFSHT). Näheres hierzu in der Inline-Beschreibung der Enter-Datei.

Geöffnete SYSOUT-/SYSLST-Dateien, die in einem Element einer PLAM-Bibliothek abgelegt werden, können von der User Option OPEN nicht erfaßt werden.
Ab SYSFILE V13.0A (OSD 4.0) steht das Kommando **MOD**IFY**-SYSF**ILE**-OPT**IONS **LIB**RARY**-OUT**PUT**-COPY**=**Y**ES zur Verfügung. Damit wird der Inhalt eines Sysfile PLAM-Elements zusätzlich in eine temporäre Datei auf Platte geschrieben. Zum Zeitpunkt, wo das PLAM-Element geschlossen wird, wird auch die temporäre Sysfile-Datei gelöscht. Die temporäre Sysfile-Datei wird von der User Option OPEN angezeigt. Der Inhalt der Datei kann mit dem Action-Code D sichtbar gemacht werden.

Bei Anwendung des Action-Codes D auf Dateien mit einem angezeigten LASTP-Wert > 200 kann das Übertragen aller bereits beschriebenen PAM-Blöcke in die Hilfsdatei eine lange Zeit in Anspruch nehmen. Der Benutzer wird daher gefragt, ob er an der Anzeige des Inhalts der vollständigen Datei interessiert ist, oder nur die letzten *n* PAM-Blöcke anschauen möchte:

```
nnnnn Pam-blocks will be copied before the complete contents of file
:pvs:$user-id.filename   is available for display.
View the complete file or the first/last n Pam-blocks only ?
(DUE/+n/n/K1/?)
```

Eingabe Enter bzw. DUE1-Taste: Kompletten Dateiinhalt anzeigen.
n : Nur die letzten *n* geschriebenen Pam-Blöcke der Datei anzeigen.
+*n* : Nur die ersten *n* Pam-Blöcke der Datei anzeigen.
K1-Taste: Abbruch; keine Aktion.

Ignorieren der Schutzattribute beim Löschen von Dateien / Jobvariablen

EI

ERASE... ,IGNORE=(ACCESS,EXDATE,RDPASS,WRPASS)
Beim Löschen werden die Dateiattribute ACCESS=READ und RETPD>0, die Basic Access Control List sowie evtl. vorhandene Read- und Write-Passworte ignoriert.

Passworte für Datei / Jobvariable anzeigen

FP

Bei einer Datei wird das Kommando FSTAT *datei*,P ausgeführt. Damit werden alle vorhandenen Dateipassworte ausgegeben. Das gleiche gilt auch für Jobvariablen. Der Action-Code FP ist dem Systemverwalter unter TSOS vorbehalten.

FTA

File-Transfer Activate partner. Es wird ein Kommando MODIFY-FT-PARTNER ..., STATE=*ACTIV für den mit dem Action-Code FTA markierten Partnerrechner ausgeführt. Der Action-Code kann nur unter TSOS und in einer mit NSTAT selektierten Dateienliste eingegeben werden.

Alle Tasks anzeigen, die eine Datei eröffnet haben

OPN

Es werden alle Tasks (TSN's) angezeigt, die die mit dem Action-Code bezeichnete Datei gegenwärtig im Zugriff haben (User Option: `OPEN IOENPRSEC,S`).
Falls es sich bei der Datei um eine im Output-Modus eröffnete SAM-Datei handelt, können deren aktuelle Inhalte im Anschluß an OPN mit dem Action-Code D (Display) sichtbar gemacht werden. Für weitergehende Informationen siehe Seite A574.
Falls es sich bei der Datei um eine PLAM-Bibliothek handelt, werden alle Tasks angezeigt, die diese Bibliothek geöffnet haben, sowie die geöffneten Elemente dieser Bibliothek. Der Action-Code D ist auf Elemente nicht möglich.

Alle Tasks anzeigen, die eine Monitor Jobvariable im Zugriff haben

OPN

Es werden alle Tasks (TSN's) angezeigt, die die mit dem Action-Code bezeichnete Jobvariable gegenwärtig per ONEVT-Kommando überwachen (User Option ONEVT).

Passworte für passwortgeschützte Dateien/Bibl.Elemente/Jobvariablen setzen

SP

Für die mit diesem Action-Code bezeichneten Dateien / PLAM-Bibliothekselemente / Jobvariablen werden die zum uneingeschränkten Zugriff notwendigen PASSWORD-Kommandos abgesetzt.

Der Action-Code SP ist auch in BS2000-Systemen mit Passwortverschlüsselung anwendbar (ENCRYPT=YES). Das Absetzen der Passworte für geschützte PLAM-Bibliothekselemente ist ab LMS Version 3.0 möglich.

SPREL

Für die mit diesem Action-Code bezeichneten Dateien / PLAM-Bibliothekselemente / Jobvariablen werden die mit SP bekannt gemachten Passworte wieder aus der Passwort Tabelle entfernt (PASSWORD ...,REL=Y).

Anzeigen der Anzahl der unbenutzten Datenblöcke bei ISAM-Dateien

SFI

Es wird das BS2000-Kommando /SHOW-FREE-ISAM-SPACE für die angegebene ISAM-Datei ausgeführt.
Der Action-Code kann nur unter der Benutzerkennung TSOS ausgeführt werden. Außerdem muß das Softwareprodukt SPACEOPT V2.0 oder höher der Firma FSC installiert sein.

Kommandos

```
 dd.mm.yy   hh:mm:ss    HOST: ........   TSN: ....     SELECTED USER-ID: ........
 COMMAND :
 LASTP TYP FILENAME                                      AGE PASS LACC  ACTION
    30 *I  xxxxxxxxxxxx  . . . . . . . . . . . . . . .  143 *      62 :
    77 *P  xxxxxxxxxxxx  . . . . . . . . . . . . . . .  143 *      62 :
     9 *P  xxxxxxxxxx  . . . . . . . . . . . . . . . .  143 *      62 :
    49 *P  xxx . . . . . . . . . . . . . . . . . . . .    4 . . .   2 :
    61 *I  xxxxxxxxxxxxxxxx  . . . . . . . . . . . . .    4 . . .   2 :
    10 *S  xxxxxxxxxxxxxxxx  . . . . . . . . . . . . .    4 . . .   4 :
    73  S  xxxxxxxxxxxxxxxxxxx . . . . . . . . . . . .    4 . . .   2 :
    29  S  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .    4 . . .   2 :
     0  N  xxxxxxxxxxxxxxxxxxxxx . . . . . . . . . . .      . . .   0 :
   245 *P  xxxxxxxxxxx . . . . . . . . . . . . . . . .    4 . . .   2 :
    49  P  xxxxxx  . . . . . . . . . . . . . . . . . .    2 . . .   0 :
   237 *P  xxxxxxxxxxxxx . . . . . . . . . . . . . . .    4 . . .   4 :
    12  S  xxxxxxxxxxxxxxx . . . . . . . . . . . . . .    8 . . .   7 :
     1  S  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .   45 . . .   4 :
     1  S  xxxxxxxxxxxxxxxxxxxx  . . .  . . . . . . . .  42 . . .   4 :
    14 *P  xxxxx . . . . . . . . . . . . . . . . . . .  290   RW   62 :
     9 *P  xxxxxxxxxxxxxx  . . . . . . . . . . . . . .  148    W   53 :
     9  P  xxxxxxxxxxxxxxxxx . . . . . . . . . . . . .  148 *       1 :
 *TAPE  S  xxxxxxxxxxx . . . . . . . . . . . . . . . .    4 . . .   4 :
    11 *P  xxxxxxxxxxxxxxxxxxxxxxx . . . . . . . . . .  148 *ERW   62 :
 List continues  (P=1/T=187/H=0)          For Help:  ? / ?cmd / ?all / ??
```

File-Transfer unter TSOS (FT/ONXFT)

FT

Falls CFS unter der Benutzerkennung TSOS aufgerufen wurde, so sind bei dem Kommando FT bzw. bei der Variablen Action ONXFT folgende Besonderheiten zu beachten:

Werden Dateien übertragen, die nicht unter der Kennung TSOS gespeichert sind und wird im Maskenfeld "Filename in Target System" *SAME eingetragen, so werden die übertragenen Dateien nicht unter TSOS, sondern unter der originalen Benutzerkennung eingerichtet. Dies gilt sowohl für die Transfer-Direction TO, als auch für die Transfer-Direction FROM.

Werden Dateien auf einen anderen Rechner übertragen, die im eigenen System mit einem READ-Passwort belegt sind, so wird dieses im NCOPY-Kommando von CFS automatisch versorgt, sofern vom Benutzer im Maskenfeld "Password for remote File" noch kein Passwort eingetragen wurde. Bei Systemen mit Passwortverschlüsselung muß das lokale READ-Passwort vom Benutzer selbst in der FT-Maske angegeben werden.

BS2000-Volume (Public-/Privat-Platte) im Display-Modus anzeigen

I *vsn*/*device*

CFS bietet, ähnlich wie das Dienstprogramm DPAGE, die Möglichkeit, eine ganze Magnetplatte im Display-Modus anzuschauen. Hierzu ist mit dem Kommando I *vsn*/*device* in der Dateienliste ein Eintrag für die gewünschte Magnetplatte zu erzeugen. Auf diesen Eintrag kann anschließend der Action-Code D (Display) bzw. M (Modify) angewendet werden.

Join Entry Services (JES)

JES [*params*] Das System JES zur Verwaltung von Joineinträgen wird aufgerufen.
Die wahlweise Angabe von Parametern bewirkt die Auswahl von Joineinträgen. Dem Benutzer wird in der Joinliste das Ergebnis der Selektion präsentiert. Ohne Angabe von Parametern wird die Selektionsmaske von JES ausgegeben.
Das Programm JES ist in einem eigenen Dokument "JES-Benutzerhandbuch" beschrieben.

Beispiele:

`JES`
Es wird die Komponente JES aufgerufen und dem Benutzer die Selektionsmaske zur Auswahl der Joineinträge angeboten. Falls der Benutzer zuvor JES über das Kommando CFS verlassen hatte, verzweigt das Kommando JES an die zuletzt verlassene Stelle innerhalb von JES.

`JES;`
Es wird in JES verzweigt und dem Benutzer sofort die Liste aller Joineinträge der online verfügbaren Pubsets präsentiert. Durch das Zeichen ';' wird das Absenden der leeren Selektionsmaske von JES simuliert.

`JES TSOS,Q`
Es wird in JES verzweigt. Der Parameter TSOS,Q bezeichnet eine Selektionsbedingung, die sofort ausgeführt wird. Dem Benutzer werden als Ergebnis der Selektion die Joineinträge der Kennung TSOS auf allen verfügbaren Pubsets angezeigt.

Join Entry Services Group (JESG)

JESG [*params*] Das System JESG zur Benutzergruppenverwaltung wird aufgerufen.
Die wahlweise Angabe von Parametern bewirkt die Auswahl von Gruppeneinträgen. Dem Benutzer wird in der Benutzergruppenliste das Ergebnis der Selektion präsentiert. Ohne Angabe von Parametern wird die Selektionsmaske von JESG ausgegeben.

Das Programm JESG ist in einem eigenen Dokument "JESG-Benutzerhandbuch" beschrieben.

Beispiele:

`JESG`
Es wird die Komponente JESG aufgerufen und dem Benutzer die Selektionsmaske zur Auswahl der Gruppeneinträge angeboten. Falls der Benutzer zuvor JESG über das Kommando CFS verlassen hatte, verzweigt das Kommando JESG an die zuletzt verlassene Stelle innerhalb der Gruppenverwaltung.

`JESG;`
Es wird in JESG verzweigt und dem Benutzer sofort die Liste aller Joineinträge aller Gruppeneinträge präsentiert. Durch das Zeichen ';' wird das Absenden der leeren Selektionsmaske von JESG simuliert.

MRS Pubsetverwaltung

MRS Es werden alle Einträge des MRS-Katalogs in einer Maske angezeigt.
Mit dem Action-Code VOL werden alle Volumes eines PVS bzw. Volume-Sets in einer weiteren Maske angezeigt.
Mit dem Action-Code FREE werden alle freien Speicherblöcke (Extents) auf einem Pubset oder Volume, sortiert in absteigender Reihenfolge nach der Größe, angezeigt (FREE Extent Liste).
In der FREE Extent Liste kann mit dem Action-Code FIL eine Dummy-Datei in einen oder mehrere freie Extents gelegt werden.

Für eine ausführliche Beschreibung der MRS-Verwaltung wird auf das Benutzerhandbuch TAS (Task Services) verwiesen.

Speicherbelegung der Public-Platten anzeigen (PUBSP)

PUBSP [*pubset*] [, FILE=*datei*] [, SUM | SAT | CAT] [, SLAVE= Y|N]

Es wird eine Maske ausgegeben, in der die Speicherbelegung aller Volumes von allen online verfügbaren Pubsets oder eines bestimmten Pubsets angezeigt wird. Falls die Funktion PUBSP auch Benutzern unter Kennungen ungleich TSOS zur Verfügung stehen soll, so muß der Systemverwalter unter TSOS über einen Holdertask das Zusatzprodukt TASHT gestartet haben (/ENTER CFS.S. LMSLIB(TASHT)).

pubset Es wird die Speicherbelegung des angegebenen Pubsets angezeigt. Die Pubset-Bezeichnung muß ohne Doppelpunkte angegeben werden, z.B. PUBSP A oder PUBSP TEST.

datei Die Speicherbelegungstabelle wird in einer Datei mit dem angegebenen Namen festgehalten. Die Ausgabe der Bildschirmmaske wird in diesem Fall unterdrückt.

SUM Es wird nur die Summenzeile (siehe unten) für jedes Public-Volume-Set ausgegeben.

SAT Es wird nur die Summenzeile (siehe unten) für jedes Public-Volume-Set ausgegeben. Anstelle der Spalte #Files wird eine Spalte PagesNextSat ausgegeben. Die entsprechenden Werte zeigen für jedes Pubset die Anzahl Seiten bis zum nächsten Saturation Level an.

CAT Das Format der Ausgabeliste wird insofern geändert, als anstelle der Spalten #Files, Device, MN die Anzahl der benutzten und freien Katalogblöcke (4K-Blöcke) angezeigt wird. Mit dieser Option kann sich der Systemverwalter einen Überblick über evtl. drohende Katalogengpässe bei den einzelnen Pubsets verschaffen.

SLAVE=Y|N Im Standardfall werden in der PUBSP-Liste auch die Pubsets angezeigt, die im Slave-Modus an aktuellen Host importiert sind.
Mit SLAVE=N werden in der PUBSP-Liste nur die Pubsets angezeigt, für die der aktuelle Host auch der Master ist.

Die PUBSP-Maske besitzt folgendes Format:

```
 dd.mm.yy    hh:mm:ss            PUBSPACE - LIST                          TSN: ....
 COMMAND :
 PVS  VSN             Used-Pages           Free-Pages     #Files Device   MN
 I    PUBI00         337833  (74%)        113772  (26%)       485 D348E   D0    NK2
 J    PUBJ00         279081  (61%)        172524  (39%)       869 D348E   D1
 J    PUBJ01         276282  (61%)        175323  (39%)       857 D348E   D2
 J    total          555363  (61%)        347847  (39%)      1726
 K    PUBK00         375108  (83%)         76479  (17%)       485 D348E   D9        PAG
 L    PUBL00         364137  (80%)         87468  (20%)      1531 D348E   BA    NK2
 M    PUBM00         363582  (80%)         88023  (20%)       773 D348E   BB    NK2
 N    PUBN00         324099  (71%)        127506  (29%)       944 D348E   D7    NK2
 O    PUBO00         311967  (69%)        139638  (31)       1229 D348E   B8    NK2  PAG
 O    PUBO01         309450  (68%)        142155  (32%)      1232 D348E   B9    NK2  PAG
 O    PUBO02         307782  (68%)        143823  (32%)      1234 D348E   B7    NK2  PAG
 O    total          929199  (68%)        425616  (32%)      3695
 P    PUBP00         392952  (87%)         58653  (13%)      1482 D348E   E0
 2    PUB200         285999  (63%)        165606  (37%)      1417 D348E   E7
 4    PUB400         392436  (86%)         59169  (14%)      2384 D348E   E6
 SUM:  5870865       4320708  (73%)       1550157  (27%)     14922      Return: K1
```

Im Kommandofeld der PUBSP-Maske können folgende Kommandos angegeben werden:

/cmd — beliebiges BS2000-Kommando

n/OC*n*/CC*n*/DC*n*/CD
: beliebiges Connection bezogenes Kommando. Eine vollständige Beschreibung der einzelnen Kommandos finden Sie im Kapitel 9 "Connections" ab Seite 337.

+ | - | +*n* | -*n* | ++ | --
: Positionieren innerhalb der Liste

S,*such* — Es wird der nächste Eintrag in der Liste gesucht, der die in der Suchbedingung angegebenen Zeichenfolgen enthält. Für *such* kann eine einfache oder eine komplexe Suchbedingung angegeben werden.

HC *datei* | NHC — Ein-/Ausschalten des Hardcopy-Modus zur Protokollierung der Masken

PUBSP [*pubset*] — weiteres PUBSP-Kommando für ein bestimmtes Pubset

NP* — Aktualisieren der letzten Pubspace-Liste

LST — Verlassen der PUBSP-Maske und Rückkehr in die Dateienliste von CFS bzw. JES (gleiche Wirkung wie K1, siehe unten)

F3 — Protokollierung der aktuellen Maske in eine Standard Hardcopy-Datei

K1 — Verlassen der PUBSP-Maske und Rückkehr in die Dateienliste von CFS

Speicherbelegung der Privatplatten anzeigen (PRIVSP)

PRIVSP [*priv*] [, FILE=*datei*]

Es wird eine Maske ausgegeben, in der die Speicherbelegung aller online verfügbaren Privatplatten angezeigt wird. Falls die Funktion PRIVSP auch Benutzern unter Kennungen ungleich TSOS zur Verfügung stehen soll, so muß der Systemverwalter unter TSOS über einen Holdertask das Zusatzprodukt TASHT gestartet haben (/ENTER CFS. S.LMSLIB(TASHT)).

priv Es wird die Speicherbelegung der Privatplatten angezeigt, deren VSNs mit der angegebenen Teilqualifizierung beginnen. Beispiel: PRIVSP PVT.

datei Die Speicherbelegungstabelle wird in einer Datei mit dem angegebenen Namen festgehalten. Die Ausgabe der Bildschirmmaske wird in diesem Fall unterdrückt.

Im Kommandofeld der PRIVSP-Maske können folgende Kommandos angegeben werden:

/cmd beliebiges BS2000-Kommando

n/OC*n*/CC*n*/DC*n*/CD
beliebiges Connection bezogenes Kommando. Eine vollständige Beschreibung der einzelnen Kommandos finden Sie im Kapitel 9 "Connections" ab Seite 337.

+ | - | +*n* | -*n* | ++ | --
Positionieren innerhalb der Liste

HC *datei* | NHC Ein-/Ausschalten des Hardcopy-Modus zur Protokollierung der Masken

PRIVSP [*priv*] weiteres PRIVSP-Kommando mit einer anderen Teilqualifizierung

NP* Aktualisieren der letzten Privsp-Liste

LST Verlassen der PRIVSP-Maske und Rückkehr in die Dateienliste von CFS bzw. JES (gleiche Wirkung wie K1, siehe unten)

F3 Protokollierung der aktuellen Maske in eine Standard Hardcopy-Datei

K1 Verlassen der PRIVSP-Maske und Rückkehr in die Dateienliste von CFS

Dateiattribute und Passworte wiederherstellen

RECAT *datei* [, CATID=*pvs*] [, UID=*user-id*] [, CJ] [, IMP] [, NC] [, NU] [, **P**] [, **KD** [,RESET=PLAM] [,BLKC]]

Das Kommando RECAT setzt eine mit Hilfe der Variablen Action ONXLISTCAT *datei* erzeugte Katalogauszugsdatei voraus. Für jeden in der Listcat-Datei gesicherten Katalogeintrag wird geprüft:

a) existiert die Datei ?

b) wenn ja: Stimmen die im Katalogauszug festgehaltenen Dateiattribute wie SHARE=YES/NO, ACCESS=READ/WRITE mit den aktuellen Attributen der datei überein ? Falls Nein, wird ein /CATALOG-Kommando abgesetzt, um die Dateiattribute wieder auf die ursprünglichen Werte zu setzen.

Bei Dateien werden die folgenden Attribute geprüft und ggf. wieder hergestellt:

- SHARE=YES
- ACCESS=READ
- RETPD
- AUDIT
- BACKUP-Level
- LARGE=YES
- DESTROY=YES

- BASIC-ACL
- GUARDS

Bei Job-Variablen werden folgende Attribute geprüft und ggf. wieder hergestellt:

- SHARE=YES
- ACCESS=READ
- RETPD
- BASIC-ACL

CATID=*pvs* Die zu bearbeitenden Dateien/Jobvariablen werden auf dem angegebenen Pubset gesucht. Dies erfolgt unabhängig vom der in der Listcat-Datei aufgezeichneten Catalog-ID der Datenobjekte.
pvs ist ohne Doppelpunkte anzugeben, z.B. `CATID=TEST`

UID=*user-id* Die zu bearbeitenden Dateien/Jobvariablen werden unter der angegebenen Benutzerkennung gesucht. Dies erfolgt unabhängig vom der in der Listcat-Datei aufgezeichneten Benutzerkennung der Datenobjekte.
user-id ist ohne $-Zeichen und Punkte anzugeben, z.B. `UID=TT08`

Die Parameter CJ, IMP, NC und NU des RECAT-Kommandos sind im CFS-Benutzerhandbuch auf Seite **267** beschrieben.

KD Keep Date. Es wird das in der Listcat-Datei aufgezeichnete Creation- und Change Date für die zu bearbeitenden Dateien wieder hergestellt. Die KD-Option ist unwirksam für Jobvariablen und für Dateien auf einer Privatplatte.
Bei eingeschaltetem (KLA) Modus wird zusätzlich auch das Last Access Datum wieder hergestellt.

RESET=PLAM Ein eventuell vorhandener Indikator für eine PLAM-Bibliothek wird im Katalog zurückgesetzt. Dieser Parameter ist nur in Verbindung mit KD erlaubt.

BLKC Bei Banddateien wird der Block-Count im Katalog auf den in der Listcat-Datei vorgegebenen Wert gesetzt. Dieser Parameter ist nur in Verbindung mit KD erlaubt.

P Bei Angabe der P-Option (RECAT *datei*,P) werden auch Paßworte (READ/WRITE/EXEC) wieder hergestellt, sofern diese zum aktuellen Zeitpunkt nicht mehr bestehen oder nicht mehr mit den festgehaltenen Inhalten übereinstimmen. Auch verschlüsselte Passworte werden wieder hergestellt.

Hinweis:

Die P-Option kann nur angegeben werden, falls CFS unter der Kennung TSOS geladen wurde. Die Listcat-Datei muß ebenfalls unter TSOS erstellt worden sein (ONXLISTCAT *datei*).

Sysout/Syslst-Datei von laufendem Task umschalten

SW SYSOUT | SYSLST[*nn*]= *datei* | *bibl*(*typ*/*element*[/*vers*]) | *NEXT | *LIB-ELEM(LIB=...) [,E]
,TSN=*tsn* | ,MONJV=*jobvar* [,MSG=N|Y] [,INF=N|Y[(*sw*)]|A[(*sw*)]]

Das Kommando SW schaltet für einen laufenden Batchtask die aktuelle Sysout- bzw. Syslst-Datei auf eine neue Datei um. Die bisherige Datei wird geschlossen und steht danach für Auswertungs- bzw. Archivierungszwecke zur Verfügung.

SYSOUT= Eine zugewiesene Sysout-Datei wird umgeschaltet.

SYSLST[*nn*]= Eine zugewiesene Syslst- bzw. Syslst01 - 99 Datei wird umgeschaltet.

datei Neuer Dateiname für Sysout/Syslst. Falls keine Benutzerkennung angegeben ist, wird die User-Id des laufenden Batchtasks angenommen.

bibl(*typ*/*element*[/*vers*]) | *LIB-ELEM(LIB=*bibl*,ELEM=*element*,TYPE=*typ*)

Die Sysout/Syslst-Datei wird in ein PLAM-Bibliothekselement umgeleitet. Der Name des Bibliothekselements kann in allgemeiner CFS oder SDF-Syntax angegeben werden.

*NEXT Ausgehend vom bisherigen Dateinamen wird der neue Dateiname für SYSFILE automatisch erzeugt. Falls der aktuelle Dateiname mit einer Laufnummer .nnn (.000 - .999) endet wird diese um 1 erhöht. Bei Überlauf wird mit der Nummer .000 fortgesetzt.
Falls der Dateiname nicht mit .nnn endet, wird die Suffix .001 angehängt.
Falls die aktuelle SYSOUT/SYSLST-Datei einem PLAM-Bibliothekselement zugewiesen ist, wird bei *NEXT im Elementnamen die Suffix .001 angehängt bzw. um 1 erhöht.

E Extend.
Die neue Datei bzw. das Bibliothekselement wird mit Open=Extend eröffnet.
Standardmäßig wird das neue Datenobjekt mit Open=Output eröffnet.

TSN=*tsn* Die Umschaltung erfolgt für den Batchtask mit der angegebenen TSN.

MONJV=*jobvar* Die Umschaltung erfolgt für den Batchtask, der angegebenen Monitor-Jobvariablen verknüpft ist.

MSG=Y In der aktuellen SYSOUT/SYSLST-Datei wird der Umschaltvorgang durch Ausgabe einer Meldung der Form '`switch initiated by TSN=...: ASSIGN-SYSxxx TO=yyyyy`' angekündigt.
Standard: MSG=N

INF=<u>N</u> Die Umschaltung erfolgt ohne Rückmeldung. Nicht erfolgreiches Umschalten kann damit nicht mittels CFS-Prozedursprache abgefragt oder in der Meldungszeile von CFS angezeigt werden.

INF=Y[(*sw*)] Die Umschaltung erfolgt mit <u>synchroner</u> Rückmeldung. Dies bedeutet, daß CFS solange wartet, bis die Umschaltung erfolgt ist bzw. mit einer Fehlermeldung abgewiesen wurde. Die Umschaltung im Zieltask kann erst erfolgen, sobald dieser aktive Kontrolle über die CPU erhält.

Zur Synchronisierung wird im CFS-Task standardmäßig der Prozessschalter 30 verwendet. Falls dieser Schalter für andere Zwecke benötigt wird, kann auch ein beliebiger anderer Schalter *sw* zwischen 5 und 30 angegeben werden.

Über die Prozedursprache von CFS kann der Schalter 31 bezüglich der erfolgreichen Ausführung des Kommandos abgefragt werden: SETSW ON=31 weist auf einen Fehler in der Ausführung des Kommandos hin.

INF=A[(*sw*)] Die Umschaltung erfolgt mit <u>asynchroner</u> Rückmeldung. Dies bedeutet, daß CFS sofort nach Absetzen des Kommandos weiterarbeitet. Jedoch ist Schalter 30 bzw. *sw* auf ON gesetzt. Erst sobald Schalter 30 auf OFF zurückgesetzt ist, kann im Schalter 31 der Erfolg der Operation abgefragt werden.

Bibliotheken / Dateien reorganisieren

REORG *bibl-1* [, *bibl-2*] [,[*:x:*] [,*smspar*] | *vsn/device* | PAD=*nn* | NR | DF[O] | SPACE=(*xxxx,yyy*) | **KD**]

Die angegebene Bibliothek/Datei wird reorganisiert, indem alle Elemente/Datensätze in eine neue Bibliothek/Datei übertragen werden. Für eine ausführliche Beschreibung aller Parameter des Kommandos REORG wird auf Seite 268 verwiesen.

KD

Keep Date. Die reorganisierte Datei/Bibliothek erhält das gleiche Change Date wie die Ursprungsdatei und nicht das Datum des heutigen Tages, wie dies ohne Angabe der KD-Option der Fall wäre. Die KD-Option ist unwirksam, falls die Datei/ Bibliothek auf eine Privatplatte reorganisiert wird.
Die KD-Option bewirkt außerdem, daß die Backup Version # und die Save Version # der Datei beibehalten werden. Dies ist für den Archive-Sicherungsstatus von Bedeutung.
Bei eingeschaltetem KLA-Modus (Kommando (KLA)) wird zusätzlich das Last Access Date der Eingabe- und Ausgabe-Datei wieder hergestellt.

Task Services (TAS)

TAS [*params*]

Die Task Services (TAS) werden aufgerufen. Die wahlweise Angabe von Parametern bewirkt die Auswahl von Tasks. Dem Benutzer wird in der Taskliste das Ergebnis der Selektion präsentiert. Ohne Angabe von Parametern wird die Selektionsmaske des Programms TAS ausgegeben.

Das Programm TAS ist in einem eigenen Dokument "TAS-Benutzerhandbuch" beschrieben.

Hinweis:

Komponente TAS auch für nichtprivilegierte (Nicht-TSOS) Anwender. Ein Normalbenutzer kann einen durch die Einstellung /OPTION TESTPRIV=(x,y) abgestuften Funktionsumfang von TAS (Task Services) nutzen. Voraussetzung hierfür ist, daß unter TSOS der TAS-Holdertask gestartet wurde: /ENTER $CFS.CFS.S.LMSLIB (TASHT). Der Benutzer ruft wie unter TSOS das Kommando TAS auf. Durch Absenden der leeren Selektionsmaske erhält er alle Tasks mit LOGON unter seiner Kennung (TESTPRIV=(1,1)) bzw. alle Tasks im System (TESTPRIV=(8,1)). Mit (8,1) erhält der Benutzer lesenderweise auch alle übrigen Informationen, die TAS dem Systemverwalter unter TSOS zur Verfügung stellt. Mit (8,8) ist auch der modifizierende Zugriff möglich. TASHT ist kostenpflichtig (einmaliges Entgelt), steht aber in der ausgelieferten CFSLIB als uneingeschränkt lauffähige Testversion zur Verfügung.

Beispiele:

```
TAS;
```
Es wird in TAS verzweigt und dem Benutzer sofort die Liste aller zur Zeit aktiven Tasks präsentiert. Durch das Zeichen ';' wird das Absenden der leeren Selektionsmaske von TAS simuliert.

```
TAS TSOS
```
Es wird in TAS verzweigt. Der String TSOS wird im ersten Feld (USER-ID) der Selektionsmaske eingetragen und die entsprechende Selektion ausgeführt. Als Ergebnis wird dem Benutzer eine Liste aller zur Zeit aktiven Tasks, die unter TSOS angeloggt sind präsentiert.

TASNO;DSSM
Es wird in TAS verzweigt und es erfolgt keine Auswahl von Tasks (TASNO). Im Feld "COMMAND" der zweiten TAS-Maske wird das Kommando DSSM ausgeführt. Damit werden alle Subsysteme und deren Status ausgegeben.

Last Access Date unverändert lassen nach lesenden Dateizugriffen

(KLA) Der Keep Last Access Modus wird durch das Kommando (KLA) eingeschaltet und ist bis zur Eingabe des Kommandos (KLA)OFF wirksam.

Der KLA-Modus bewirkt:

1) Nach dem lesenden Dateizugriff durch die Variable Action ONXFIND, die User Option FIND, sowie durch den Action-Code D (Display) wird das alte Last Access Date wieder hergestellt.

2) Bei den Variablen Actions ONXCOPY, ONXMOVE, ONXREORG mit der Option KD (Keep Date) wird zusätzlich zum Change Date auch das alte Last Access Date der Eingabe- und Ausgabedatei wieder hergestellt.

3) Bei der Anzeige der Liste von Elementen einer PLAM-Bibliothek wird trotz des lesenden Zugriffs auf die Bibliothek das ursprüngliche Last Access Date beibehalten.
Beim lesenden Zugriff auf einzelne PLAM-Elemente (Variable Action ONXFIND.., Action-Code D(isplay)) wird auch für diese Elemente das ursprüngliche Last Access Date beibehalten.

Das Kommando (KLA) ist eine spezielle Anwendung für die optionale Open-/Close-Exit Routine von CFS. Durch diesen Exit kann der Systemverwalter mit selbstgeschriebenen Routinen den Zugriff auf sensible Dateien verhindern bzw. kontrollieren.

Aufrufkonventionen und Parameterliste des Open-/Close-Exits

Die Adresse des Open-/Close-Exits ist in den Feldern YOPENAD/YCLOSEAD im Global-Bereich (Makro/DSECT GLOBAL) einzutragen (z.B. über CFSMAIN).
Der Open- bzw. Close Exit wird von CFS jeweils vor und nach dem DVS-Makroaufruf OPEN/CLOSE angesprungen.

Registerbelegung / Parameter:

R1 Opcode.
X'00000001': Exit-Aufruf vor der Ausführung des DVS-Makros.
Close-Exit: Register R0 enthält die Adresse des noch geöffneten FCB.
X'00000002': Exit-Aufruf nach der Ausführung des DVS-Makros.
Open-Exit: Register R0 enthält die Adresse des geöffneten FCB.

R13 Adresse einer 18 Worte umfassenden Save-Area.

R14 Rückkehradresse in CFS.

R15 Einsprungadresse: Beginn des Exits.

Rückkehrinformationen:

Rückkehrinformationen im Register 15 werden nur nach Rücksprung aus dem Open-Exit mit Opcode X'00000001' ausgewertet.

R15 = X'00000000' DVS Open-Aufruf darf ausgeführt werden.
R15 > X'00000000' DVS Open-Aufruf darf nicht ausgeführt werden.

CFS-Holdertask, TSOS-Dienste für nichtprivilegierte Benutzer

Durch die Enter-Prozedur J/CFSHT in der PLAM-Bibliothek CFS.S.LMSLIB kann unter der Kennung TSOS der CFS-Holdertask gestartet werden. Über diesen Holdertask können nichtprivilegierten CFS-Benutzern bestimmte Dienste angeboten werden, die sonst nur TSOS-Benutzern zur Verfügung stehen. CFSHT realisiert die folgenden Dienste:

1) User Option OPEN zur Auswahl der geöffneten Dateien.
   Es werden dem nichtprivilegierten Benutzer alle Dateien angezeigt, die von Tasks unter der Kennung des Benutzers eröffnet wurden. Der Benutzer kann mit dem Parameter D-IO auch die IO-Statistik für die unter seiner Kennung eröffneten Dateien aktivieren.
   Der Holdertask ignoriert Parametereingaben des Benutzers, die zur Anzeige von geöffneten Dateien führen würden, auf die er keinen Zugriff hat, z.B. die Angabe einer fremden Benutzerkennung oder TSN.

2) Action-Code OPN. Es werden alle TSN's angezeigt, die die mit OPN markierte Datei eröffnet haben. Es können dabei auch TSN's unter anderen Kennungen als der des Benutzers angezeigt werden. Mit dem Action-Code S (STA *tsn*) werden die Status-Informationen der Tasks angezeigt. Der Action-Code S für Tasks, die nicht unter der eigenen Benutzerkennung laufen, kann über CFSMAIN ausgeschaltet werden.

3) Action-Code D (Display) zum Sichtbarmachen des Inhalts von geöffneten Ausgabedateien. Diese Funktion ist nur in einer mit der User Option OPEN selektierten Dateienliste möglich.
   Der CFS-Holdertask erzeugt eine Hilfsdatei mit dem bisher geschriebenen Inhalt der geöffneten SAM-/PAM-Datei. Diese Hilfsdatei wird in der Kennung des Benutzers angelegt. Der nichtprivilegierte CFS löscht die Hilfsdatei beim Verlassen der Display-Funktion.

4) :*cmd* im Kommandofeld der Dateienliste. Damit kann der nichtprivilegierte Benutzer bestimmte, vom Systemverwalter vorgegebene BS2000-Kommandos unter der Kennung TSOS zur Ausführung bringen. Der Output (SYSOUT) des Kommandos wird dem Benutzer wieder zugestellt. Im SYSOUT-Protokoll des Holdertasks erfolgt eine Protokollierung aller ausgeführten und zurückgewiesenen Kommandos zusammen mit TSN und User-ID des Requestors.

   In den folgenden Abschnitten wird beschreiben, welche Möglichkeiten der Systemverwalter hat, um einzelnen Benutzern (Benutzerkennungen) bestimmte BS2000-Kommandos freizuschalten.

   Damit die Kommandoausführung für nichtprivilegierte Benutzer überhaupt aktiv wird, muß im CFS-Holdertask eine Steuerdatei mit dem Linknamen CFSHTCMD zugewiesen werden, z.B. FILE CFS.HTCMD,LINK=CFSHTCMD. Die Standard-SAM Datei CFS.HTCMD enthält neben beliebig vielen Kommentarsätzen (erste Stelle '*', aber nicht *ALL), Musterkommandos und User-Id Informationen.

   Alle Änderungen an der Datei CFS.HTCMD können vom Systemverwalter im laufenden Betrieb vorgenommen werden. Die Änderungen werden sofort wirksam.

   Beispiel:

   ```
   STAJV
   SETJV
   $TEST
   SHOW-DAB
   VERIFY
   JOIN TEST,PASS=
   ENTER $TEST.CFS.E.BATCH
   ```

$TESTUTM
START-SS 'UTM','AIDSYS'
$SYSPRIV
*ALL

Die Kommandos SHOW-DAB, VERIFY, ENTER $TEST.CFS.E.BATCH sind für Benutzer unter der Kennung $TEST reserviert.
Das Kommando START-SS mit den Parametern UTM oder AID darf nur unter der Kennung $TESTUTM ausgeführt werden.
Unter der Kennung SYSPRIV können mit :*cmd* alle BS2000-Kommandos ausgeführt werden. Die Kommandos VERIFY und REPAIR-DISK-FILE, die vom Benutzer u.U. eine zusätzliche Antwort verlangen, werden vom CFS-Holdertask ohne Benutzerrückfrage ausgeführt.
Unter allen nicht explizit angegebenen Kennungen, also unter allen Kennungen ungleich $TEST, $TESTUTM und $SYSPRIV sind die am Anfang aufgeführten Kommandos STAJV und SETJV erlaubt. Diesen Kommandos ist kein $user-id Satz vorangestellt.

Bestimmte Kommandos für Benutzer zulassen:
Die angeforderten Kommandos werden vom Benutzer im Kommandofeld von CFS in der Form :*cmd* [*param*] eingegeben.
Die CFS.HTCMD-Datei enthält Mustereingaben der Form *xxx* [*yyyyy*]
CFSHT prüft, ob für den Benutzer eines der Musterkommandos *xxx* mit dem angegebenen Kommando *cmd* übereinstimmt. Paßt kein Musterkommando auf die Benutzereingabe, so erhält der Anwender die Meldung "CFSHT: Command not allowed for user" und das Kommando wird nicht ausgeführt.

Falls in der Mustereingabe nur der Kommandoname angegeben ist, kann der Benutzer das Kommando mit beliebigen Parametern ergänzen. Der Kommandoname muß in der gleichen Weise wie in der Mustereingabe eingegeben werden. Andere, in SDF auch mögliche Schreibweisen des Kommandos werden zurückgewiesen.

Kommandoparameter in der Mustereingabe:
Falls in der Mustereingabe neben dem Kommandonamen auch Parameter angegeben sind, werden diese von CFSHT gesondert geprüft. Die Parameter in der Mustereingabe werden als CFS-Suchanweisung behandelt, falls sie mit ' , -' , oder (beginnen. Ist dies nicht der Fall, so wird die angegebene Zeichenfolge über eine einfache CLC-Schleife in den vom Benutzer eingegebenen Parametern gesucht.

Bei einer CFS-Suchbedingung können komplexe UND-/ODER-/Wildcard- bzw. negative Verknüpfungen angegeben werden. Der Nachteil besteht darin, daß Hochkommas im Suchstring verdoppelt werden müssen.

Bei einer einfachen Suchbedingung kann nur ein zu suchender String angegeben werden. Die Zeichenfolge wird dafür so gesucht, wie in der HTCMD-Datei angegeben. Es ist in diesem Fall keine Verdoppelung von Hochkommas notwendig.

Beispiel 1:
JOIN TEST,PASS=C'A
Die Parameter des Kommandos JOIN werden nach der Zeichenfolge TEST, PASS=C'A durchsucht.

Beispiel 2:
JOIN ('TEST','UTM')*'PASS=C''A'
Die Parameter des Kommandos JOIN werden nach TEST oder UTM und dahinter nach PASS=C'A durchsucht.

Alle Kommandos für Benutzer zulassen bis auf wenige Ausnahmen:
Die CFS.HTCMD-Datei enthält Mustereingaben der Form *-xxx*
CFSHT prüft, ob für den Benutzer eines der Musterkommandos *xxx* mit dem angegebenen Kommando *cmd* übereinstimmt. Paßt ein Musterkommando auf die Benutzereingabe, so wird dieses zurückgewiesen und der Anwender erhält die Meldung "CFSHT: Command not allowed for user". Alle Kommandos außer den mit *-xxx* aufgeführten werden für den Benutzer zugelassen.

MVS PO-Bibliotheken in LMS-Bibliotheken importieren (MVSIMP)

(MVSIMP)

Mit dem Kommando (MVSIMP) können PO (partitioned organized) Bibliotheken aus dem Betriebssystem MVS in LMS-Bibliotheken des BS2000 übertragen werden. Der Inhalt der PO-Bibliothek muß auf MVS-Seite mit dem Dienstprogramm **IEBUPDTE oder IEBPTPCH** (Anweisung PUNCH) auf ein Magnetband exportiert werden.

IEBUPDTE: SYSUT2: RECFM=FB, LRECL=80, BLKSIZE=2000
IEBPTPCH: SYSUT2: RECFM=FB, LRECL=81, BLKSIZE=2025

Im BS2000 muß das Band mit

```
/FILE datei,LINK=MVSIMP,VOLUME=...,DEVICE=..., STATE=FOREIGN
```

zugewiesen werden. Das Kommando (MVSIMP) überträgt die auf dem Magnetband befindlichen Bibliothekselemente in eine LMS-Bibliothek mit dem festen Namen LMS. MVSIMPORT.

CFS bietet auch eine Funktion TRANSF zum Überspielen von BS2000-Dateien/Bibliotheken auf Magnetbänder, die im MVS mit Standard-Dienstprogrammen übernommen werden können. Das System TRANSF ist in einem eigenen Dokument beschrieben.

USERLIB.BEISPIEL

/FILE CFS.STARTUP.BEISPIEL,LINK=CFSCMD

Die CFS-Startup-Datei stellt eine spezielle CFS-Prozedur dar, deren Anweisungen beim Aufruf des Programms automatisch abgearbeitet werden. In der Startup-Datei steht der volle Funktionsumfang von CFS zur Verfügung.
Die ausgelieferte Datei soll einige Standardanwendungsfälle aufzeigen. Die Datei kann durch Anpassung der Parameteranweisungen jederzeit auf die eigenen Bedürfnisse angepaßt werden.

Sollte die Datei CFS.STARTUP.BEISPIEL in der CFS-Installationskennung nicht existieren, so ist diese auch als Element S/CFS.STARTUP.BEISPIEL in der Bibliothek CFS.USERLIB. BEISPIEL enthalten und kann aus dieser Bibliothek selektiert werden.

Beispiele für Standardaktionen in einer Startup-Datei:

- P-Tasten laden
- CFS-Parametereinstellungen verändern (UO, ERT, CP, CSO, ...)
- CFS-Kommandogedächtnis laden
- Benutzerumgebung herstellen (USERLIB, USERACT, ITEM-File, MENUE-File, ...)
- Connections eröffnen
- Termine prüfen
- Umschalten auf die Tastatur bis zum normalen Ende von CFS
- Kommandogedächtnis sicherstellen
- Eröffnete Connections automatisch schließen

Die ausgelieferte Startup-Datei stellt folgende Dateizuordnungen als Voraussetzung zum Testen der in dieser Beschreibung vorgestellten Kommandoprozeduren her:

CFS.USERLIB.BEISPIEL	User-Library für CFS-Kommandoprozeduren
CFS.USERACT.BEISPIEL	Definitionsdatei für benutzereigene Action Codes

CFS.USERACT.BEISPIEL

Mit den benutzereigenen Action Codes wird dem Administrator und jedem einzelnen Benutzer die Möglichkeit geboten, sich individuelle Arbeitsabläufe über %*act*, zusätzlich zu den standardmäßig angebotenen Action Codes, bereitzustellen. Jeder dieser benutzereigenen Action Codes kann mit einem CFS-Kommando, einer CFS-Kommandoprozedur oder einer BS2000-Prozedur verknüpft werden. Die nachfolgend beschriebenen Action-Codes werden in der Datei CFS.USERACT.BEISPIEL definiert.

Sollte die Datei CFS.USERACT.BEISPIEL in der CFS-Installationskennung nicht existieren, so ist diese auch als Element S/CFS.USERACT.BEISPIEL in der Bibliothek CFS.USERLIB. BEISPIEL enthalten und kann aus dieser Bibliothek selektiert werden.

Die Datei CFS.USERACT.BEISPIEL muß zum Testen der nachfolgend beschriebenen Action Codes mit LINK=USERACT und die Bibliothek CFS.USERLIB.BEISPIEL als aktuelle USERLIB zugewiesen sein (wird in CFS.STARTUP.BEISPIEL durchgeführt). Ist dies der Fall, so stehen folgende Action Codes zur Verfügung:

Aufruf von CFS-Kommandos

%VSN	Anzeige des Allocated Space mit Größe der Extents über alle VSN's
%FC	Umsetzung einer Datei mit dem CFS-Kommando FC
%REO	Reorganisieren einer Datei/Bibliothek mit Ausweichplatte und Defragment
%FT	Filetransfer mit FT-BS2000 und vordefinierten Berechtigungen
%EDR	Datei mit EDOR bearbeiten
%PRL	Auswahl aller TSN's in TAS, die einen Print-Lock auf die markierte Datei halten

Aufruf von BS2000-Kommandos

%MFA	Katalogattribute mit SDF-Kommando /MOD-FILE-ATT auf bestimmte Werte setzen
%CAT	Katalogattribute mit ISP-Kommando /CAT auf bestimmte Werte setzen
%SEC	Datei mit /SECURE-RESOURCE-ALLOC exclusiv reservieren
%PR	Datei mit bestimmten Print-Parametern ausdrucken
%CI	Sekundärschlüssel für NK-ISAM Datei erzeugen (mit SDF-Dialog)

Starten von BS2000-DO-Prozeduren mit Parametern im Enter

%ASS	Assembler-Übersetzung mit Überwachung im Enter starten (Job-Report)
%COB	Cobol-Übersetzung mit Überwachung im Enter starten (Job-Report)

Kombinierte Verwendung von Kommandos und Action Codes durch Aufruf von CFS-Prozeduren

%P0	Katalogattribute ändern mit aktualisieren des Tabelleneintrags
%P1	Katalogattribute ändern mit aktualisieren des Tabelleneintrags
%U	FSTAT VTOC=YES und aktualisieren des Tabelleneintrags
%DUPK	Prüfen, ob in einer ISAM-Datei doppelte Keys vorhanden sind

%P0, %P1 -> DO CFS.USERLIB.BEISPIEL(**PROTECT**):
BS2000-Kommando absetzten, auf entsprechenden Eintrag positionieren, Action Code U eintragen, auf ursprünglichen Eintrag zurückpositionieren.

%U -> DO CFS.USERLIB.BEISPIEL(**UPDATE-VTOC**):
BS2000-Kommando FSTAT ...,VTOC=YES für Datei absetzen, auf entsprechenden Eintrag positionieren, Action Code U eintragen, auf ursprünglichen Eintrag in der Dateienliste zurück-positionieren.

%DUPK -> DO CFS.USERLIB.BEISPIEL(**DUPKEY**):
Datei mit User Options KEYPOS, KEYLEN, RECFORM auswählen, verzweigen in CFS-Editor, satzweises Vergleichen der ISAM-Keys über Wegschreiben in Prozedurvariablen.

Verzweigen in den CFS-Editor über CFS-Kommandoprozeduren

%DF	Entkomprimierte Anzeige von mit FLAM komprimierten Dateien im Display.
%DP	Anzeige von mit SHARUPD eröffneten NK-ISAM Dateien.

%DF -> DO CFS.USERLIB.BEISPIEL(**DISPLAY-FLAM**):
auf markierten Eintrag positionieren, Action Code D eintragen, DA (Data Access) FLAM aktivieren, umschalten auf Tastatur bis CFS-Editor beendet wird, danach auf ursprünglichen Eintrag in der Dateienliste zurückpositionieren.

%DP -> DO CFS.USERLIB.BEISPIEL(**DISPLAY-NKISAM**):
auf entsprechenden Eintrag positionieren, Action Code D eintragen, falls notwendig SR-Modus (Shared Update) aktivieren, falls mit NKISAM-POOL gearbeitet wird: POOL-Namen aus OPEN-List oder über Prompting anfordern, Kommando /CREATE-ISAM-POOL und /ADD-ISAM-POOL-LINK absetzen, umschalten auf Tastatur bis CFS-Editor beendet wird, danach auf ursprünglichen Eintrag in der Dateienliste zurückpositionieren.

Bearbeiten der **Access-Control-List** (SECOS) über CFS-Kommandoprozeduren

%A	ACL anzeigen
%AC	ACL kopieren von einer Datei, die zuvor mit Action-Code * markiert wurde
%AE	ACL löschen
%AEU	ACL User-Id entfernen
%AEG	ACL Group-Id entfernen
%AU	User-Id in ACL aufnehmen
%AG	Group-Id in ACL aufnehmen
%AX	mehrere User-Ids/Group-Ids in ACL aufnehmen
%AM	ACL modifizieren
%AMU	Rechte für User-Id modifizieren
%AMG	Rechte für Group-Id modifizieren
%AMO	Rechte für Others modifizieren

-> DO CFS.USERLIB.BEISPIEL(**ADD-ACL** / **MOD-ACL** / **REM-ACL**):
Zum Anzeigen und Verändern der Access-Control-List stehen diese beiden Kommandoprozeduren zur Verfügung. In den Prozeduren werden die notwendigen BS2000-Kommandos in der Form abgesetzt, daß die wichtigsten Parameter schon vorbelegt sind und die fehlenden Informationen im SDF-Dialog vervollständigt werden müssen. Am Ende der Prozeduren wird der jeweilige Eintrag in der Dateienliste aktualisiert.

DO LIB

Die CFS-Prozedur bietet die Möglichkeit, über Kurzcodes, die als Parameter beim Prozeduraufruf mitgegeben werden, häufig benötigte Standardselektionen auszuführen. Je nach Parameter wird das Inhalts-verzeichnis einer Bibliothek angezeigt. Über Stellungsparameter kann ein Kurzcode für die Bibliothek, eine Suchbedingung bezüglich des Elementnamens, eine Typbezeichnung und eine Altersangabe als Auswahlparameter übergeben werden.

Die CFS-Prozedur kann durch Anpassung der Kurzcodes und Bibliotheksnamen auf die eigene Arbeitsumgebung angepaßt werden.

Aufruf:

DO LIB [, *code*] [, *suchanw*] [, *typ*] [, *age*]

	Ohne Angabe von Parametern werden alle Elemente aus einer in der Prozedur definierten Standardbibliothek ausgewählt.
code	Es werden alle Elemente der durch *code* definierten Bibliothek ausgewählt (evtl. auch nur Elemente eines bestimmten Typs). Wurde ein ungültiger Kurzcode angegeben, so wird als Bibliothek die Standardbibliothek angenommen und die Stellungsparameter logisch um eine Position nach links verschoben. *code* wird in diesem Fall als Such-anweisung bezüglich des Elementnamens interpretiert.
suchanw	Es werden alle Elemente der durch *code* definierten Bibliothek ausgewählt, in deren Elementnamen der String *suchanw* vorkommt.
typ	Es werden alle Elemente vom Elementtyp *typ* aus der entsprechenden Bibliothek ausgewählt.
age	Es werden alle Elemente mit AGE=*age* aus der entsprechenden Bibliothek ausgewählt.

Beispiele:

DO LIB	Alle Elemente aus der Standardbibliothek (CFS.S.LMSLIB)
DO LIB,U	Alle Elemente der Bibliothek CFS.USERLIB.BEISPIEL
DO LIB,C	Alle Elemente der Bibliothek CFSLIB
DO LIB,U,CONN	Alle Elemente der Bibliothek CFS.USERLIB.BEISPIEL, in deren Elementnamen der String 'CONN' enthalten ist
DO LIB,S,,J	Alle Elemente der Bibliothek CFS.S.LMSLIB vom Elementtyp J

DO COMP-LIB

Die CFS-Prozedur bietet die Möglichkeit, die Inhaltsverzeichnisse zweier Bibliotheken darauf hin zu überprüfen, welche Elemente gemeinsam vorhanden, exclusiv vorhanden und neueren Datums sind. Über ein Auswahlmenü können die gewünschten Elemente ausgewählt werden.

Aufruf:

DO COMP-LIB [, LIB1=*library*] [, LIB2=*library*] [, TMPIND=*tmp*]

Ohne Angabe von Parametern wird eine Maske ausgegeben, in der die Bibliotheksnamen und der Tempfile-Indikator angefordert werden.

LIB1=*library* Name der ersten Bibliothek

LIB2=*library* Name der zweiten Bibliothek

TMPIND=*tmp* Einleitungszeichen für temporäre Dateien (STD: #)

DO COMP-PVS

Die CFS-Prozedur bietet die Möglichkeit, die katalogisierten Dateien zweier Pubsets oder User-Ids darauf hin zu überprüfen, welche Dateien gemeinsam vorhanden, exclusiv vorhanden und neueren Datums sind. Über ein Auswahlmenü können die gewünschten Dateien ausgewählt werden.

Aufruf:

DO COMP-PVS [, PVS1=*pvs1*] [, UID1=*user-id1*] [, PVS2=*pvs2*] [, UID2=*user-id2*] [, TMPIND=*tmp*]

Ohne Angabe von Parametern wird eine Maske ausgegeben, in der die Werte für die PVS-Id, User-Id und der Tempfile-Indikator angefordert werden.

PVS1=*pvs1* Public-Volume-Set der ersten Benutzerkennung

UID1=*user-id1* Name der ersten Benutzerkennung

PVS2=*pvs2* Public-Volume-Set der zweiten Benutzerkennung

UID2=*user-id2* Name der zweiten Benutzerkennung

TMPIND=*tmp* Einleitungszeichen für temporäre Dateien (STD: #)

DO DUPKEY

Die CFS-Prozedur prüft, ob in einer vorgegebenen ISAM-Datei doppelte Schlüssel enthalten sind und gibt diese am Bildschirm aus. Am Ende wird die Gesamtzahl der gefundenen doppelten ISAM-Schlüssel am Bildschirm ausgegeben. Die für die Verarbeitung notwendigen Informationen KEYPOS, KEYLEN und RECFORM werden in der Prozedur ermittelt.

Aufruf:

DO DUPKEY [, *datei*] [, OUT=N]

Ohne Angabe von Parametern wird der Dateiname und der Indikator, ob doppelte Schlüssel am Bildschirm ausgegeben werden sollen, in einer Maske angefordert.

datei Es wird für die angegebene Datei geprüft, ob doppelte Schlüssel vorhanden sind. Die Ausgabe der Parametermaske entfällt.

OUT=N Die Anzeige der doppelten Schlüssel wird unterdrückt.

DO WRITE-MN

Die CFS-Prozedur schreibt bestimmte Datensätze der aktuell angezeigten Display-Datei in eine zweite Datei. Über Parameter kann eine CFS-Suchbedingung für die Treffersätze übergeben werden. Außerdem ist es möglich auch Datensätze im Umfeld der Treffersätze mit wegzuschreiben (*m* Sätze vor und *n* Sätze nach dem Treffer).

Aufruf:

DO WRITE-MN [, ITEM=*such*] [, WFILE=*datei*] [, M=*m*] [, N=*n*]

Ohne Angabe von Parametern wird eine Maske ausgegeben, in der die Suchanweisung, der Name der Ausgabedatei und die Anzahl Datensätze vor und nach dem Treffer angefordert werden.

ITEM=*such* — Datensätze, in denen die durch *such* spezifizierte CFS-Suchbedingung erfüllt ist, werden als Treffersätze behandelt und weggeschrieben.

WFILE=*datei* — Die gewünschten Datensätze werden in die Ausgabedatei geschrieben.

M=*m* — Zusätzlich zum Treffersatz werden die letzten *m* Sätze vor dem Treffer in die Ausgabedatei geschrieben (Standard: 0).

N=*n* — Zusätzlich zum Treffersatz werden *n* Sätze nach dem Treffer in die Ausgabedatei geschrieben (Standard: 0).

DO WRITE-BIS-TREFFER

Die CFS-Prozedur schreibt alle Sätze ab der aktuellen Anzeigeposition in der Display-Datei bis zu einem vorgegebenen Treffer in eine zweite Datei. Über Parameter kann die Suchbedingung für den Treffer angegeben werden.

Aufruf:

DO WRITE-BIS-TREFFER [, ITEM=*such*] [, WFILE=*datei*]

Ohne Angabe von Parametern wird eine Maske ausgegeben, in der die CFS-Suchanweisung und der Name der Ausgabedatei angefordert werden.

ITEM=*such* — Der erste Datensatz, bei dem die durch *such* spezifizierte Suchbedingung erfüllt ist, wird als Treffersatz behandelt. Das Übertragen von Sätzen in die Ausgabedatei wird bei diesem satz beendet.

WFILE=*datei* — Die Datensätze bis zum Treffer werden in die Ausgabedatei geschrieben.

DO WRITE-PART

Die CFS-Prozedur sucht in der aktuell angezeigten Display-Datei nach einem vorgegebenen String und schreibt ab dem Suchstring *n* Bytes als Recform=V Sätze in eine Datei.

Aufruf:

DO WRITE-PART [, ITEM=*string*] [, WFILE=*datei*] [, LEN=*n*]

Ohne Angabe von Parametern wird eine Maske ausgegeben, in der der Suchstring, der Name der Ausgabedatei und die Länge der zu erzeugenden Datensätze angefordert werden.

ITEM=*string* Ab dem Suchstring *string* wird jeweils ein eigener Datensatz in der vorgegebenen Länge geschrieben.

WFILE=*datei* Die gewünschten Teile der Datensätze werden in die Datei *datei* geschrieben.

LEN=*n* Ab dem Suchbegriff werden *n* Bytes als Datensatz weggeschrieben.

DO DEL-STRING

Die CFS-Prozedur entfernt abhängig von einem vorgegebenen Suchbegriff bestimmte Teile aus den Datensätzen. Über Parameter kann der Suchbegriff, der Offset zum Suchbegriff, ab dem gelöscht werden soll und die Länge des zu löschenden Teils übergeben werden.

Die CFS-Prozedur setzt voraus, daß die zu bearbeitende Datei entweder eine ISAM-Datei ist oder mittels des Kommandos IS bereits im ISAM-Modus angezeigt wird.

Durch modifizieren der Verarbeitungsanweisung (S1,'...'='',C) kann die Prozedur auch zum Einfügen bzw. Ersetzen von Strings verwendet werden.

Aufruf:

DO DEL-STRING [, ITEM=*string*] [, OFFSET=*offs*] [, LEN=*n*]

Ohne Angabe von Parametern wird eine Maske ausgegeben, in der der Suchstring, der Offset zum Suchbegriff und die Länge des zu löschenden Teils angefordert werden.

ITEM=*string* — Suchstring *string* in dessen Umfeld vorgegebene Datenbytes entfernt werden sollen.

OFFSET=*offs* — Distanz *offs* zum Anfang des Suchstrings, ab dem in der gewünschten Länge gelöscht werden soll (Standard: 0).

LEN=*n* — Ab dem vorgegebenen Suchbegriff und Offset werden *n* Datenbytes aus den Datensätzen entfernt.

DO INS-HEADER

Diese Prozedur soll zeigen, daß neben dem Löschen, Einfügen oder Ersetzen von Strings auch die Möglichkeit besteht, in bestimmten Situationen auch Datensätze in eine bestehende Datei einzufügen. In einer Testdatei werden bestimmte Satzinhalte dahingehend geprüft, daß bei einem Wechsel der Prüffelder neue Datensätze eingefügt werden (z.B. Gruppenwechsel, Überschriften, ...).

DO OC

Die CFS-Prozedur eröffnet je nach Parameterübergabe Connections zu bestimmten vordefinierten Anwendungen bzw. bei $DIALOG, in bestimmte Benutzerkennungen. Die Prozedur stellt eine flexible Alternative zum CFS-Menüsystem für Connections (automatisierter Aufbau über vordefinierte Verbindungen) dar. Die Prozedur kann beliebig erweitert werden und muß an die individuelle Umgebung angepaßt werden (User-Id, Account, Passwort, UCON-Berechtigung, KDCSIGN, ...).

Aufruf:

DO OC [, *id*] Es wird eine Connection in die gleiche Benutzerkennung (gleiche Abrechnungsnummer, gleicher Host) eröffnet, unter der der CFS-Grundtask läuft. Ein vorhandenes Logon-Passwort wird über BS2000-Prompting angefordert.

id Über den Stellungsparameter *id* kann eine bestimmte vordefinierte Verbindung aufgebaut werden. Der übergebene String *id* wird als Label interpretiert und es werden die in der Prozedur an dieser Stelle hinterlegten OC-Anweisungen zur Ausführung gebracht.

Beispiele:

DO OC,TSOS Anwendung $DIALOG, Kennung $TSOS, gleicher HOST
DO OC,TSOS11 Anwendung $DIALOG, Kennung $TSOS, Rechner HOST11
DO OC,KURS Anwendung $DIALOG, Kennung $KURS, gleicher HOST
DO OC,UTMTEST Anwendung UTMTEST mit automatischer Anmeldung
DO OC,CONS Anwendung $CONSOLE, gleicher HOST
DO OC,CONS11 Anwendung $CONSOLE, Rechner HOST11

DO UTM-OC

Die CFS-Prozedur stellt Ihnen eine Verbindung zur UTM-Anwendung TEST zur Verfügung. Falls schon eine Connection zu dieser Anwendung existiert, wird in diese Verbindung verzweigt. Ansonsten ermittelt die Prozedur die nächste freie Connection, baut die Verbindung auf und setzt die hinterlegten Eingaben (z.B. KDCSIGN, RES datei, ...) automatisch ab.

Die gleiche Funktionalität steht auch in der Prozedur OC zur Verfügung (DO OC,UTMTEST).

Die nachfolgenden Prozeduren stellen eine Verbindung zu $CONSOLE zur Verfügung. Falls schon eine Connection zu $CONSOLE besteht, wird in diese Connection verzweigt (Continue in Connection). Existiert die Konsolverbindung noch nicht, so wird diese aus der Prozedur, gemäß den hinterlegten Berechtigungen, aufgebaut und anschließend die Kontrolle an die Tastatur übergeben. Sobald der Benutzer die Konsolverbindung verläßt, wird diese automatisch abgebaut.

DO CONSOLE-OC

Die CFS-Prozedur stellt eine Konsolverbindung über eine generierte Nebenkonsole mit allen Eingabemöglichkeiten zur Verfügung.

Vor Aufruf der Prozedur muß die UCON-Userid und das Passwort angepaßt werden.

DO CONSOLE-OC-DYN

Die CFS-Prozedur stellt eine dynamische, nicht vorgenerierte Konsolverbindung zur Verfügung, für die eine hinterlegte Operator-Role zugewiesen wird. Diese Variante des Verbindungsaufbaus steht ab BS2000 OSD/V1.0 zur Verfügung.

Vor Aufruf der Prozedur muß die User-Id, das Passwort und die Operator-Role angepaßt werden.

DO CONSOLE-OHNE-INPUT

Die CFS-Prozedur stellt eine Verbindung zu $CONSOLE zur Verfügung, wobei die Meldungen am Bildschirm ausgegeben werden, der Benutzer jedoch keine Eingabemöglichkeit erhält.
Die Prozedur kann durch zweimaliges Betätigen der K2-Taste beendet werden.

Vor Aufruf der Prozedur muß die UCON-Userid und das Passwort angepaßt werden.

DO VMCONS-OC

Die CFS-Prozedur baut eine Verbindung zu $VMCONS (Virtuelle Maschine VM2000) auf, wobei ein VM-Dialog gemäß den hinterlegten Eingaben zur Verfügung gestellt wird.
Sobald der Benutzer die VM-Konsole verläßt, wird diese automatisch wieder abgebaut.

Vor Aufruf der Prozedur muß eventuell das für $VMCONS definierte Passwort angepaßt werden.

/ENTER CFS.USERLIB.BEISPIEL(AUTOMATIC-CONSOLE)

In dieser BS2000-Prozedur wird CFS aufgerufen und eine Verbindung zu $CONSOLE eröffnet. Auf bestimmte Meldungen, die in dieser Konsolverbindung ausgegeben werden, können beliebige Reaktionen festgelegt werden.

- ausstehende Anfragen können automatisch beantwortet werden.
- von Benutzern unter bestimmten Kennungen können über TYPE Kommandos übergeben werden, die an der Konsole ausgeführt werden.

Vor Start der Prozedur muß die UCON-Userid und das Passwort angepaßt werden. Außerdem sollten, um den Aufwand möglichst gering zu halten, nicht benötigte Meldungsschlüssel mit /ASR für die UCON-Berechtigung entfernt werden.

Im Bibliothekselement CFS.USERLIB.BEISPIEL(D/KONSOLMELDUNGEN) sind die Meldungen hinterlegt, auf die die Prozedur reagieren soll.

Beispiel:

```
'NKVT012 PREMOUNT TAPE'+:1-20:'?' .MOUNT
'SYSTEMDUMP DESIRED'+:1-20:'?' .SYSDUMP
'CFSCON:' .CFSCONSOLE
'TERMINATE CONSOLE' .EXIT
```

Jeder Datensatz dieses Bibliothekselements besteht aus einer CFS-Suchanweisung bezüglich des Inhalts der Konsolmeldung und einem SKIP-Label.
Ist eine Suchanweisung erfüllt, d.h. eine Treffermeldung wurde ausgegeben, so wird das hinterlegte SKIP-Label (.label) in eine CFS-interne Variable mit dem Namen &SLABEL eingetragen und in der Prozedur zur Verarbeitung an die entsprechende Stelle verzweigt.

DO PROCESS-LIST

Die CFS-Prozedur stellt einen Rahmen für die Bearbeitung aller in der aktuellen Dateienliste sichtbaren Einträge dar (mit Action Code - ausgeblendete Datenobjekte werden nicht bearbeitet). Als Verarbeitung stehen alle CFS-Funktionen (Kommandos, Action-Codes, Variable Actions) zur Verfügung.
Zur Demonstration wurde als Verarbeitung das Austauschen von Strings gleicher Länge in den Dateiinhalten gewählt (ohne mit FIND-REWR zu arbeiten).

Die CFS-Prozedur kann durch modifizieren der Verarbeitungsanweisungen für individuelle Anwendungsfälle angepaßt werden.

Aufruf:

DO PROCESS-LIST [, OLD=*str1* , NEW=*str2*]

Ohne Angabe von Parametern werden der Such- und der Ersetzstring in einer Maske ange-fordert.

OLD=*str1* , NEW=*str2*

Es wird in allen Sätzen der in der Liste sichtbaren Datenobjekte nach dem String *str1* gesucht und dieser durch den String *str2* ersetzt. Wichtig ist, daß Such- und Ersetzstring die gleiche Länge haben, da CFS standardmäßig nicht mit Ein-/Ausfügen arbeitet (nur möglich im ISAM-Modus!).

Falls die Bearbeitung für ein Datenobjekt fehlerhaft ist, wird eine entsprechende Fehlermeldung mit dem DMS-Errorcode ausgegeben und die Verarbeitung mit dem nächsten Datenobjekt fortgesetzt.

DO TERMIN

Die CFS-Prozedur bietet die Möglichkeit, über das CFS-Terminverwaltungssystem auf eine vorgegebene Termindatei zuzugreifen. Als Aktionen stehen die Anzeige aller offenen Termine, aller bzw. ausgewählter Termine und die Neuaufnahme von Terminen zu einem beliebigen Datum zur Verfügung.

Die Prozedur verwendet standardmäßig die Termindatei CFS.ITEM.*jobname*.

Über Parameter können die Aktion (Anzeige, Neuaufnahme), der Zugriff auf offene/alle Termine, ein Suchstring für den Termintext sowie Datum und Uhrzeit für die Neuaufnahme beeinflußt werden.

Aufruf:

DO TERMIN Es werden alle offenen Termine aus der zugewiesenen Termindatei angezeigt.

DO TERMIN , *datum* [, ...]

datum Es werden alle offenen Termine aus der zugewiesenen Termindatei für den Tag *datum* angezeigt. Für *datum* kann auch +*n* (alle Termine von heute bis heute+*n* Tage) angegeben werden.

TEXT=*string* Es werden alle offenen Termine aus der zugewiesenen Termindatei angezeigt, die im Termintext den String *string* enthalten. *string* steht für eine einfache oder mehrfache Suchanweisung.

ALL=Y Es werden alle (inklusive der bereits erledigten STATE=OK) Termine aus der zugewiesenen Termindatei angezeigt.

DO TERMIN , ADD [, ...]

ADD Es wird eine Maske zur Neuaufnahme eines Termins für den aktuellen Tag mit der aktuellen Uhrzeit ausgegeben. (z.B. Aufnahme von Kurzinfos).

datum [, *uhrzeit*] Es wird eine Maske zur Neuaufnahme eines Termins für den durch *datum* festgelegten Tag und der durch *uhrzeit* festgelegten Zeit ausgegeben.

DO LIST

Die CFS-Prozedur bietet die Möglichkeit, sich die Datensätze von bestimmten Datenobjekten druckaufbereitet in eine Datei ausgeben zu lassen. Am Anfang der Ausgabedatei wird ein Inhaltsverzeichnis erzeugt.

In der Prozedur wird eine Variable Action (ONXLIST) definiert. Der Benutzer wird aufgefordert, die zu bearbeitenden Datenobjekte mit dem Action Code X zu markieren. Sobald der Benutzer das Ende der Dateienliste einmal quittiert hat bzw. die Ausführung der Variablen Action durch das Kommando A forciert hat, wird die Ausgabedatei erzeugt, die Terminierungsabfrage durch die Prozedur beantwortet und anschließend die Ausgabedatei mit dem Action Code PD auf einem vordefinierten Drucker ausgegeben.

Aufruf:

DO LIST [, *datei*]	Es werden alle mit X markierten Datenobjekte in einer Ausgabedatei (Standard: #LIST.TOC) aufgelistet und anschließend ausgedruckt.
datei	Name der Ausgabedatei.

DO RUN-DATE

Die CFS-Prozedur ist eine Demonstration für die Funktionalität des RUN-Moduls DATE, das in CFS-Prozeduren allgemeine Prüfungen und Berechnungen in Verbindung mit einem bestimmten Datum zur Verfügung stellt.

Die *RUN (DATE) Anweisung greift zur Analyse und Berechnung von Datumsangaben automatisch auch auf eine eventuell vorhandene Datei CFS.IFREE zu, in der zusätzliche arbeitsfreie Tage, regionale Feiertage usw. hinterlegt werden können (siehe Terminverwaltung).

Die angebotenen Datumsfunktionen können zur Plausibilitätsprüfung von Datumseingaben und zur terminabhängigen Verarbeitung in Jobs verwendet werden. So ist es zum Beispiel sehr einfach möglich, am Anfang eines Jobs (z.B. eines täglichen Repeat-Jobs) festzustellen, ob heute der erste Arbeitstag im Monat, der erste Arbeitstag der Woche, oder ein Feiertag ist. Abhängig vom Returncode kann ein Benutzerschalter oder eine Jobvariable verändert werden, aufgrund dessen im Job entsprechend verzweigt wird (z.B. Verarbeitung oder sofortiges Abmelden mit LOGOFF).

Es stehen eine Vielzahl von Funktionsaufrufen zur Verfügung:

VALID	gültiges Datum ?
FREE	Feiertag ?
DDD	laufender Tag des Jahres
WW	Kalenderwoche
WD	Arbeitstag ?
DW	Wochentag codiert in der Form: Mo=1, Die=2, ...
DW-G	Wochentag in deutsch
DW-E	Wochentag in englisch
+*n*	Datum plus *n* Tage
-*n*	Datum minus *n* Tage
WDN	nächster Arbeitstag
WDP	vorheriger Arbeitstag
WDF	erster Arbeitstag für das angegebene Monat
WDL	letzter Arbeitstag für das angegebene Monat
WW1	Datum des Montags für die angegebene Kalenderwoche

DO RUN-DATE-CAL

Die CFS-Prozedur erstellt für ein vorgegebenes Jahr über die *RUN (DATE) Funktionen eine Kalenderdatei mit den Informationen Datum, Wochentag, Feiertag, Arbeitstag, Kalenderwoche und laufender Tag des Jahres.

Der Name der Kalenderdatei, sowie das gewünschte Jahr wird bei Prozeduraufruf über eine Parametermaske angefordert.

DO INFO

Die CFS-Prozedur bietet die Möglichkeit, in einer vorgegebenen Datei (z.B. Telefonliste, Namensliste, Firmenliste, usw.) nach beliebigen Strings zu suchen und sich alle Treffersätze am Bildschirm ausgeben zu lassen. Falls nur ein Treffersatz vorhanden ist, wird dieser in der Systemzeile ausgegeben. Wurden mehrere Treffersätze gefunden, so werden diese mit WROUT am Bildschirm angezeigt und die Prozedur mit PLEASE ACKNOWLEDGE bis zur Quittierung angehalten. Am Ende der Prozedur wird in die ursprüngliche Dateienliste zurückgekehrt.

In der Prozedur wird die vorgegebene Datei selektiert, mit dem Action Code D in den CFS-Editor verzweigt, Inhalte der Treffersätze in Prozedurvariablen geschrieben, die Inhalte der Prozedurvariablen am Bildschirm ausgegeben und am Ende mit RL wieder in die ursprüngliche Dateienliste zurückgekehrt.

Die Prozedur verwendet standardmäßig eine Testdatei aus der CFS.USERLIB.BEISPIEL als Datenbasis und kann durch Änderung des Dateinamens auf die eigene Arbeitsumgebung angepaßt werden.

Aufruf:

DO INFO [, *string*] Ohne Angabe des Parameters *string* werden von allen Datensätzen der Datenbasis jeweils die ersten 80 Bytes angezeigt.

string Es werden nur Datensätze der aktuellen Datenbasis angezeigt, die den String *string* enthalten.

DO INFO-MASK

Die CFS-Prozedur bietet dieselben Möglichkeiten, wie in der vorher beschriebenen Prozedur INFO, jedoch werden die Treffersätze aufbereitet in einer FHS-Maske (mit *RDMASK) am Bildschirm angezeigt.

Vor Aufruf der Prozedur muß der Name der FHS-Bibliothek MFHSROUT eventuell im Prozedurkopf angepaßt werden (Standard: $FHS.MFHSROUT).

Aufruf:

DO INFO-MASK [, *string*] Ohne Angabe des Parameters *string* werden alle Datensätze aufbereitet angezeigt. Die Prozedur kann durch Betätigung der K1-Taste abgebrochen werden.

string Es werden nur Datensätze aufbereitet angezeigt, die den String *string* enthalten.

E CFS.USERLIB.BEISPIEL(MOVE-FILES)

Diese BS2000-ENTER-Prozedur bietet die Möglichkeit, eine vorgegebene Gruppe von Datenobjekten auf eine bestimmte Privatplatte oder ein bestimmtes Pubset zu legen (ON&MOVE).

In der Prozedur wird CFS im Basismodus aufgerufen (SETSW ON=11, *001). Im Basismodus steht nur ein Teil des Funktionsumfangs von CFS zur Verfügung (Selektion, Variable Actions, Documentation).

E CFS.USERLIB.BEISPIEL(FIND-REWR)

Diese BS2000-DO-Prozedur bietet die Möglichkeit, in einer vorgegebenen Gruppe von Datenobjekten (mehrere Bibliotheken, SAM-, ISAM- und PAM-Dateien) nach Treffersätzen mit ganz bestimmten Strings (CFS-Suchanweisung) zu suchen, diese in eine Trefferdatei zu schreiben, die Trefferdatei zu editieren (EDT oder CFS-Display) und die geänderten Datensätze wieder in die Ursprungsdatenobjekte zurückzuschreiben.

In der Prozedur wird CFS im erweiterten Prozedurmodus aufgerufen (SETSW ON=1, *002). In diesem steht der volle Funktionsumfang von CFS ohne Einschränkungen zur Verfügung.

Die Prozedur verwendet standardmäßig eine Testdatei aus der CFS.USERLIB.BEISPIEL als Datenbasis und kann durch Änderung des Dateinamens auf die eigene Arbeitsumgebung angepaßt werden.

DO FT-PAR

Die CFS-Prozedur bietet die Möglichkeit, für verschiedene Rechner individuelle Filetransferberechtigungen zu hinterlegen, die automatisch in der Filetransfermaske von CFS eingetragen werden. Die so eingetragenen Werte gelten fortan als Standard für alle FT-Aktionen, bis sie manuell oder prozedural geändert werden.

Die CFS-Prozedur kann durch Anpassung der Labels und der FT-Kommandos auf die eigene Arbeitsumgebung angepaßt werden.

Aufruf:

DO FT-PAR [, *host*] [, *direction*]

Ohne Angabe des Parameters *host* wird der in der Prozedur als Standard definierte Rechner und die dazugehörige Transferadmission (User-id, Account, Password bzw. FTAC-Transferadmission) in der FT-Maske eingetragen.

host — Es wird die für den Partnerrechner *host* hinterlegte Transferadmission (User-id, Account, Password bzw. FTAC-Transferadmission) in der FT-Maske ein-ge-tragen. Wurde ein ungültiger Host angegeben, so wird eine Fehlermeldung ausgegeben.

direction — Über den zweiten Stellungsparameter *direction* kann die Übertragungsrichtung (TO/FROM) beeinflußt werden (Standard: TO).

Neben der prozeduralen Versorgung der FT-Maske stehen noch weitere Möglichkeiten des vordefinierten Filetransfers zur Verfügung:

- Abspeichern der Eintragungen in der FT-Maske mit dem Kommando SPFT %*name* als Parameterset. Damit stehen die Eintragungen für die spätere Verwendung im FT-Kommando bzw. der Variablen Action zur Verfügung: z.B. ONXFT %HOST10

- Verwendung eines benutzereigenen Action Codes, hinter dem sich ein FT-Kommando mit Parametern für die markierte Datei verbirgt z.B. %FT aus CFS.USERACT.BEISPIEL

E CFS.USERLIB.BEISPIEL(FT-INHALT)

Die BS2000-DO-Prozedur bietet die Möglichkeit, über eine Inhaltsdatei (FSTAT-Liste, CFS-DOC-Datei, ...) mehrere Dateien von einem Partnerrechner mit FT-BS2000 abzuholen. Es wird zuerst die Inhaltsverzeichnisdatei abgeholt und anschließend über OLDLIST mit der Variablen Action ON&FT bearbeitet. Dadurch werden alle in der Inhaltsdatei aufgeführten Dateien (asynchron) zum eigenen Host übertragen.

In der Parametermaske des Job-Report muß der Name der Inhaltsdatei, der Name des Partner-rechners und die Transferadmission versorgt werden.

DO JR-PARMASK

Die CFS-Prozedur soll die verschiedenen Varianten der Parameteranforderung für CFS-Kommandoprozeduren, für BS2000-DO- bzw. S-Prozeduren über die Job-Report Parametermaske von CFS aufzeigen.

Über Texte in Doppelhochkommas können anstelle der Parameternamen beliebige Kommentare eingeblendet werden (z.B. gültige Eingabewerte für den Parameter).

Zusätzlich besteht die Möglichkeit, die Parameter dunkelgesteuert anzufordern (":D..." Secret Prompt) bzw. für lange Parametereingaben zusätzliche Eingabezeilen zur Verfügung zu stellen (":#1..." eine Zeile zusätzlich).

DO APPL

Die CFS-Prozedur verzweigt in die Komponente TAS (Task Services) und gibt den Status aller Anwendungen bzw. einer bestimmten Anwendung am Bildschirm aus. Die angezeigten Informationen umfassen die Applikationsnamen, Anzahl logischer Verbindungen, Anzahl Ein- und Ausgabenachrichten, Bytes-IN und Bytes-OUT.

Aufruf:

DO APPL [, *appl* [, CON]]   Ohne Angabe von Parametern wird eine Liste aller Anwendungen angezeigt.

appl Es wird nur die Anwendung *appl* angezeigt. Alle anderen Anwendungen werden mit dem Action Code '-' ausgeblendet.

CON Es werden die aktuell bestehenden logischen Verbindungen zur Anwendung *appl* angezeigt.

ACTION-Codes in der APPLICATION-Liste:

CON Liste aller Partner einer Applikation anzeigen.

DO UTM

Die CFS-Prozedur verzweigt in die Komponente TAS (Task Services) und gibt eine Liste aller UTM-Anwendungen bzw. einer bestimmten UTM-Anwendung am Bildschirm aus. Die angezeigten Informationen umfassen den Anwendungsnamen (Namemanager-Eintrag), Scope und die TSN der jeweiligen UTM-Tasks.

Aufruf:

DO UTM [, *anw* [, CON|D]] Ohne Angabe von Parametern wird eine Liste aller UTM-Anwendungen angezeigt.

anw Es wird nur die UTM-Anwendung *anw* angezeigt und alle anderen mit dem Action Code '-' ausgeblendet.

CON Es werden die aktuell bestehenden logischen Verbindungen zur UTM-Anwendung *anw* angezeigt.

D Es wird eine spezielle Informationsmaske für die UTM-Anwendung *anw* mit wichtigen UTM-Statusinformationen angezeigt. In dieser Maske können über die Kommandos KTA, KAA, ROOT, MPUTAREA, RSTAREA, FORMIO und FORMBR die verschiedenen UTM-Bereiche im Display-Format ausgegeben werden.

ACTION-Codes in der NAMEMANAGER-Liste UTM (Auszug):

CON Liste aller Partner der UTM-Anwendung anzeigen.

D UTM-Informationsmaske anzeigen.

DO DSSM

Die CFS-Prozedur verzweigt in die Komponente TAS (Task Services) und gibt den Status aller im Subsystem-Katalog generierten Subsysteme bzw. eines bestimmten Subsystems am Bildschirm aus. Die Informationen umfassen u.a. den Subsystem-Namen, Version, aktueller Zustand, Ladeadresse, Holder-Task, Speicherklasse und Anzahl Verbindungen zu dem Subsystem.

Aufruf:

DO DSSM [, ACT|*subsys* [, CON]]

Ohne Angabe von Parametern wird eine Liste aller generierten Subsysteme angezeigt.

ACT	Es werden nur die Subsysteme angezeigt, die sich im Zustand CREATED befinden.
subsys [, CON]	Es wird nur das Subsystem *subsys* angezeigt.
CON	Es wird eine Liste der Tasks angezeigt, die momentan mit dem Subsystem *subsys* verbunden sind.

ACTION-Codes in der DSSM-Liste (Auszug):

NP	Auswahl aller Tasks, die mit diesem Subsystem verbunden sind
L	Liste aller Module des Subsystems
CR	Subsystem aktivieren
DEL	Subsystem entladen
HOLD	Kommando HOLD-SUBSYSTEM
RES	Kommando RESUME-SUBSYSTEM

DO FT

Die CFS-Prozedur verzweigt in die Komponente TAS (Task Services) und gibt den Status aller im Netz-beschreibungsbuch von FT (File-Transfer) definierten Partnersysteme bzw. eines bestimmten Partner-systems. Die Informationen umfassen u.a. den REMSYS-Namen, SYSADR, RELADR, den aktuellen Zustand, Security-Level und Anzahl der vorhandenen Aufträge (LOC/REM).

Aufruf:

DO FT [, ACT|*partner*] Ohne Angabe von Parametern wird eine Liste aller vorhandenen FT-BS2000 Partnersysteme angezeigt.

ACT Es werden nur die FT-BS2000 Partnersysteme angezeigt, für die momentan Aufträge vorhanden sind.

partner Es wird der Zustand des angegebenen FT-BS2000 Partnersystem *partner* angezeigt.

ACTION-Codes in der FJAM-Liste:

NP Auswahl aller Filetransferaufträge mit diesem FT-BS2000 Partnersystem in CFS (NSTAT-Selektion).

DO FTAC

Die CFS-Prozedur greift im CFS-Editor direkt auf die Datei $SYSFJAM.SYSFSA zu und gibt die gewonnenen FTAC-Profil-Informationen in einer IFG-Maske am Bildschirm aus. Die Prozedur ist unter der Kennung TSOS aufzurufen.

Die angezeigten Informationen umfassen folgende Profileigenschaften:

OWNER-ID	Benutzerkennung des Eigentümers des Profils
NAME	Name des FTAC-Profils
INI	Initiator (LOCal/REMote)
DIR	Transfer-Direction (TO/FROM)
FILENAME	vorgegebener Datei-/Bibliotheksname (FILE/LIBRARY/*EXP/*EXP(LIB))
SUCC	lokales Success-Processing (NONE/YES/*EXP)
FAIL	lokales Failure-Processing (NONE/YES/*EXP)
WRI	Write-Mode (EXTend/REPlace/NEW)
IGN-LEV	Ignorieren von Security-Levels (Y/N)
EXP	Expiration-Date für Profil vorhanden (Y/N) (ab FTAC 2.0)
LCK	Profil gesperrt (Y/N) (ab FTAC 2.0)

Aufruf:

DO FTAC [, TRANS [,*suchanw*] | PARTNER] [, UID=*userid*] [, FILE=*datei*]

Ohne Angabe von Parametern wird eine Liste aller FTAC-Profile aller User-Ids mit den Standard-Informationen angezeigt.

TRANS	Es wird die TRANSFER-ADMISSION in der Liste angezeigt. Als zweiter Stellungsparameter *suchanw* kann eine Suchbedingung bezüglich der Transferadmission angegeben werden.
PARTNER	Anstelle von FAIL und WRI wird der PARTNER-Name angezeigt.
UID=*userid*	Es werden nur die Profile der Benutzerkennung *userid* angezeigt.
FILE=*datei*	Es wird eine Ausgabedatei *datei* mit den gewünschten Profil-Informationen erzeugt.

COMMAND-Eingaben in der FTAC-Maske (Auszug):

+ | -	blättern in der Liste
/*cmd*	BS2000-Kommandos absetzen
DOC *datei*	Datei mit Profilinformationen erstellen

ACTION-Eingaben in der FTAC-Maske (Auszug):

D	Profil anzeigen (SHOW-FT-PROFILE)
M	Profil modifizieren (MOD-FT-PROFILE)
P	Profil privilegieren
U	Profil die Privilegierung entziehen
L	Profil sperren (LOCK ab FTAC V2.0)
UL	Profil entsperren (UNLOCK ab FTAC V2.0)

DO PDT

Die CFS-Prozedur verzweigt in die Komponente TAS (Task Services) und gibt den Status aller generierten Geräte bzw. den Status eines bestimmten Gerätes am Bildschirm aus. Die angezeigten Informationen umfassen Volume, Device-Mnemonic, Geräte-Typ, Geräte-Zustand (ATTACHED, DETACHED), Geräteadressen und Anzahl IO's seit Startup.

Aufruf:

DO PDT [, ATT|*mn*] Ohne Angabe von Parametern wird eine Liste aller generierten Geräte angezeigt.

ATT Es werden nur die Geräte angezeigt, die sich im Zustand ATTACHED befinden.

mn Es wird nur das Gerät mit dem Mnemonic *mn* angezeigt.

ACTION-Codes in der PDT-Liste (Auszug):

NP Auswahl aller Tasks, die momentan Dateien auf diesem Volume im Zugriff haben.

DO FEP

Die CFS-Prozedur verzweigt in die Komponente TAS (Task Services) und gibt den Status aller definierten Prozessoren bzw. eines bestimmten Prozessors am Bildschirm aus. Die angezeigten Informationen umfassen Name des Prozessors, Typ, Network-Address, IP-Address, aktueller Status (BCIN, BCACT, ...) und Typ des Protokolls.

Aufruf:

DO FEP [, ACT|*proc*] Ohne Angabe von Parametern wird eine Liste aller in BCAM definierten Prozessoren angezeigt.

ACT Es werden nur Prozessoren angezeigt, die sich im Zustand BCACT befinden.

proc Es wird nur der Prozessor mit dem Namen *proc* angezeigt.

ACTION-Codes in der FEP-Liste:

NP Auswahl aller $DIALOG Tasks, die über den ausgewählten Prozessor angeschlossen sind.

DO POOL

Die CFS-Prozedur verzweigt in die Komponente TAS (Task Services) und gibt den Status aller existierenden bzw. eines bestimmten Memory Pools am Bildschirm aus. Die angezeigten Informationen umfassen u.a. den POOL-Namen, Scope, Ladeadresse, Fixed-Attribut, Anzahl angeforderter Speicherseiten und Write-Status.

Aufruf:

DO POOL [, EDT|*pool* [, PROG]]

Ohne Angabe von Parametern wird eine Liste aller existierenden Memory Pools angezeigt.

EDT	Es wird im EDT eine Liste aller existierenden Memory Pools und der jeweils verbundenen Tasks angezeigt. Dieselbe Funktionalität wird auch durch die Prozedur MEMORY-POOL-ALL realisiert.
pool [, PROG]	Es wird nur der angeforderte Memory Pool *pool* angezeigt.
PROG	Es werden alle mit dem Memory Pool *pool* verbundenen Tasks angezeigt. Als Taskinformation wird der Name des geladenen Programms ausgegeben.

ACTION-Codes in der POOL-Liste (Auszug):

D	Inhalt des entsprechenden Pools anzeigen
NP	Auswahl aller Tasks, die an diesem Pool angeschlossen sind

DO STAT-APPL-CON

Die CFS-Prozedur verzweigt in die Komponente TAS (Task Services) und ermittelt, mit welcher Anwendung ein bestimmtes Terminal verbunden ist.

Aufruf:

DO STAT-APPL-CON [, *stat*]

Ohne Angabe von Parametern wird der Stationsname zu Beginn der Prozedur über Prompting angefordert.

stat — Es wird nach logischen Verbindungen der Station *stat* gesucht und der Anwendungsname, mit dem diese Station verbunden ist, ausgegeben.

DO OMSG

Die CFS-Prozedur verzweigt in die Komponente TAS (Task Services) und zeigt eine Liste aller bzw. ausgewählter offener Konsolmeldungen an. Dazu ist innerhalb von CFS eine Connection (UCON-Verbindung) zu $CONSOLE notwendig. Die UCON-Userid und das dazugehörige Passwort müssen in der Prozedur angepaßt werden.

Aufruf:

DO OMSG [, *string*] Ohne Angabe von Parametern werden alle offenen Konsolmeldungen angezeigt.

string	Es werden nur die Konsolmeldungen angezeigt, die den Suchbegriff *string* enthalten.

ACTION-Codes in der OPEN-MESSAGE-Liste (Auszug):

FM	Anzeige der kompletten Konsolmeldung am Bildschirm (Full Message)
.	Beantwortung der Konsolmeldung mit TSN.
.*input*	Beantwortung der Konsolmeldung mit TSN.*input*

Falls eine Konsolmeldung mit einer Eingabe beantwortet werden muß, die länger als 4 Byte ist, so steht auch die Variable Action ONXCONS zur Verfügung. Damit ist es möglich, Konsolmeldungen verschiedener TSN's einheitlich mit einem beliebigen String zu beantworten. Für die TSN der jeweiligen Meldung steht bei der Definition ein Platzhalter !TSN zur Verfügung (z.B. ONXCONS !TSN.*eingabe*).

COMMAND-Eingaben in der OPEN-MESSAGE-Liste:

ATL INT=*sec*	Automatische freilaufende Ausgabe aller bzw. ganz bestimmter offener Meldungen in einem Intervall von *sec* Sekunden (Standard INT=30). Falls eine Meldung beantwortet werden soll, so ist der ATL-Modus (Automatic Task List) mit K2 unterbrechbar und kann anschließend mit ATL wieder aktiviert werden.

DO TAS

Die CFS-Prozedur verzweigt in die Komponente TAS (Task Services) und führt die beim Prozeduraufruf über einen Kurzcode angesprochene vordefinierte Taskselektion durch. Wurde beim Prozeduraufruf kein Parameter übergeben, so wird eine Übersicht aller vordefinierten Selektionen angezeigt und die gewünschte Selektion über Prompting angefordert. Nach erfolgreicher Selektion der Tasks wird automatisch in den ATL-Modus (Automatic Task List) gewechselt und die Selektion in einem Intervall von 30 Sekunden aktualisiert. Der ATL-Modus wird durch Betätigung der K2-Taste beendet.

Die Prozedur kann durch Anpassung der Kurzcodes und der TAS-Auswahlbedingungen an die eigene Arbeitsumgebung angepaßt werden.

Aufruf:

DO TAS [, *code*] Ohne Angabe von Parametern wird eine Übersicht der vordefinierten Auswahlmöglichkeiten angezeigt und eine entsprechende Auswahl durch den Benutzer erwartet:

```
----------------------------------------------------
T A S K - S E R V I C E S                    Auswahl

TAS - Selektionsmaske                        S
TAS - beenden                                E

CPU  Resourcen-Verbrauch CPU                 1
IO   Resourcen-Verbrauch IO                  2
SIZE Resourcen-Verbrauch HSP                 3
IDLE TSN's, die nicht mehr aktiv sind        4
TASKS1    Tasks unter $TT08,$KURS,$TSOS      5
SAP Tasks ($S43)                             6
RSO  RSO Prints mit Fehler                   7

Auswahl:
----------------------------------------------------
```

code Es wird die durch den *code* vordefinierte Taskselektion ausgeführt.
z.B. DO TAS,1 (Ausgabe einer Task-Hitliste bezüglich CPU-Verbrauch)

DO JES

Die CFS-Prozedur verzweigt in die Komponente JES (Join Entry Services) und führt die beim Prozeduraufruf über einen Kurzcode angesprochene vordefinierte Join-Selektion durch. Wurde beim Prozeduraufruf kein Parameter übergeben, so wird eine Übersicht der vordefinierten Selektionen angezeigt und die gewünschte Selektion über Prompting angefordert.

Die CFS-Prozedur kann durch Anpassung der Kurzcodes und der Auswahlbedingungen an die eigene Arbeitsumgebung angepaßt werden.

Aufruf:

DO JES [, *pvs*] [, *code*]

Ohne Angabe des Parameters *code* wird eine Übersicht der vordefinierten Auswahlmöglichkeiten angezeigt und eine Auswahl durch den Benutzer erwartet:

```
-----------------------------------------------------
J O I N - E N T R Y - S E R V I C E S                Auswahl

JES - Selektionsmaske                                S

aktuelle Platzbelegung auf Public                    1
aktuelle Platzbelegung auf Public/Private            2
Anzahl existierender Dateien/Jobvariablen            3
Reserven bezüglich des Pubspace-Limits               4
Reserven bezüglich des Tempspace-Limits              5
Reserven bezüglich des File-Limits                   6
Reserven bezüglich des Jobvariablen-Limits           7
Abrechnungsnummern mit TIME<100                      8
Kennungen ohne Passwort, die nicht gesperrt sind     9
Kennungen, die nicht im HOME-PVS existieren          10
SECOS: Expiration-Date für Kennung                   11
SECOS: Expiration-Date für Passwort                  12

[pvs,] Auswahl :
-----------------------------------------------------
```

pvs , *code*

Es wird die, durch den Kurzcode *code* vordefinierte Join-Selektion für das Public-Volume-Set *pvs* bzw. alle Pubsets durchgeführt.

Beispiele:

DO JES,G,1
Ausgabe einer Liste aller Kennungen des Pubsets G, absteigend sortiert nach Pubspace-Used.

DO JES,OPG,4
Ausgabe einer Liste aller Kennungen des Pubsets OPG, die Publicspeicherplatz belegen und das Pubspace-Limit nicht überschreiten dürfen (ENF=N). Als Information wird die Differenz zwischen Pubspace-Limit und Pubspace-Used zusätzlich angezeigt. Die Liste ist nach Pubspace-Differenz aufsteigend sortiert.

DO MESSUNG

Die CFS-Prozedur verzweigt in die Komponente TAS (Task Services) und ermittelt die aktuelle System-auslastung und weitere wichtige Systeminformationen (u.a. CPU-Auslastung, Disk-IO's pro Sekunde, Paging-IO's pro Sekunde, prozentuale Auslastung der Paging-Area). Aufgrund vorgegebener Grenz-werte kann veranlaßt werden, daß Meßwerte, die den Maximalwert überschreiten, bei der Ausgabe hervorgehoben (hell) dargestellt werden.
Die Prozedur kann mit der K2-Taste abgebrochen werden.

Aufruf:

DO MESSUNG,? Es wird eine Parametermaske ausgegeben, in der die individuellen Grenzwerte angepaßt werden können.

Die Parameterversorgung kann auch direkt beim Aufruf über Schlüsselworte erfolgen:

CPU=*proz1* *proz1* legt den Grenzwert für die prozentuale Gesamt-CPU-Auslastung fest.

TPR=*proz2* *proz2* legt den Grenzwert für den prozentualen TPR-Anteil an der Gesamt-CPU-Auslastung fest.

SIH=*proz3* *proz3* legt den Grenzwert für den prozentualen SIH-Anteil an der Gesamt-CPU-Auslastung fest.

DISKIO=*diskio* *diskio* legt den Grenzwert für die Anzahl Disk-IO's pro Sekunde fest.

PAGIO=*pagio* *pagio* legt den Grenzwert für die Anzahl Paging-IO's pro Sekunde fest.

PGAREA=*proz4* *proz4* legt den Grenzwert für die prozentuale Auslastung der Paging-Area fest.

TSNCPU=*cpu* *cpu* legt den taskspezifischen Grenzwert für den CPU-Verbrauch im Meß-intervall fest. Wird dieser Wert überschritten, so enthält die Ausgabeinforma-tion zusätzlich die TSN's der betroffenen Tasks (max. 3).

ACTINT=*sec1* Das Meßintervall für die TAS-Funktion SYSACT wird auf *sec1* Sekunden ge-setzt.

WAIT=*sec2* Die Wartezeit zwischen zwei Meßungen wird auf *sec2* Sekunden gesetzt.

INFO=Y Die Ausgabe der Meßwerte erfolgt nur, falls mindestens ein festgelegter Grenzwert überschritten wurde.

FILE=*datei* Zusätzlich zur Bildschirmausgabe werden die ermittelten Meßwerte auch in der Ausgabedatei *datei* abgelegt (SAM,Recform=V). Falls die Datei schon existiert, wird diese mit OPEN=EXTEND fortgeschrieben.

Beispiel für eine von der Prozedur MESSUNG erzeugte Protokolldatei:

```
--------------------------------------------------------------------------------
HOST         DATE     TIME   CPU    TU  TPR  SIH IO-DISK/S IO-PG/S PG-AREA  TSN
--------------------------------------------------------------------------------
HOST11   20.07.95 15:10:06  68.5  58.4  0.0 10.0         9       0   56%
HOST11   20.07.95 15:10:44  90.1  83.2  0.0  6.8         5       0   58%    0PV0
HOST11   20.07.95 15:11:22  91.7  85.6  0.0  6.1         7       0   59%    0PV0
HOST11   20.07.95 15:12:00  93.4  87.4  0.0  6.0         9       0   60%    0PV0
HOST11   20.07.95 15:12:35  90.5  85.6  0.0  4.9         5       0   58%    0PV0
HOST11   20.07.95 15:13:13  97.9  96.0  0.0  1.9         1       0   58%    0PV0
HOST11   20.07.95 15:13:50  86.7  83.9  0.0  2.8         5       0   56%    0PV0
HOST11   20.07.95 15:14:28  85.3  79.8  0.0  5.5         8       0   60%    0PV0
HOST11   20.07.95 15:15:06  99.0  93.8  0.0  5.2        14       0   60%    0PV0
HOST11   20.07.95 15:15:44  98.7  93.0  0.0  5.7        16       0   60%    0PV0
HOST11   20.07.95 15:16:21  97.6  94.2  0.0  3.5        14       0   60%    0PV0
HOST11   20.07.95 15:16:57  99.3  91.8  0.0  7.5        14       0   60%    0PV0
HOST11   20.07.95 15:17:34  98.9  89.5  0.0  9.4        28       0   61%    0PV0
HOST11   20.07.95 15:18:12  51.8  42.0  0.0  9.8        33       0   55%
HOST11   20.07.95 15:18:49  63.4  53.1  0.0 10.3        47       0   55%
HOST11   20.07.95 15:19:27  47.1  37.8  0.0  9.3        21       0   54%
HOST11   20.07.95 15:20:04  13.7  10.4  0.0  3.4         3       0   54%
HOST11   20.07.95 15:20:42   6.8   4.9  0.0  2.0         1       0   54%
HOST11   20.07.95 15:21:10   8.1   4.8  0.0  3.4         0       0   54%
```

DO FILEIO

Die CFS-Prozedur ermittelt zyklisch die Anzahl der IO's, die im vorgegebenen Meßintervall für jede bzw. eine bestimmte geöffnete Datei angefallen sind. Über Prozedurparameter können alle geöffneten Dateien aller bzw. eines bestimmten Tasks, eine bestimmte Datei und alle Dateien, die auf einer vorgegebenen Platte liegen ausgewählt werden.
Die angezeigten Informationen umfassen die Uhrzeit, den Dateinamen, Anzahl PAM-IO's im Meßintervall, Anzahl Read-IO's, Anzahl Write-IO's, TSN des zugreifenden Tasks und den Open-Modus der Datei.
Die ermittelten Informationen können zusätzlich in eine Datei geschrieben werden.
Die Prozedur kann mit der K2-Taste abgebrochen werden. Alternativ dazu kann eine vorgegebene Jobvariable vom Benutzer auf 'END' gesetzt werden. Die zuletzt genannte Methode garantiert stets eine definierte und ordnungsgemäße Beendigung der Prozedur.

Aufruf:

DO FILEIO Es werden alle Dateien angezeigt, für die in einem Meßintervall von 30 Sekunden mindestens eine Ein-/Ausgabe angefallen ist. Die Ausgabe ist absteigend nach IO's sortiert.

DO FILEIO ,? Es wird die Parametermaske zum ändern der vorbelegten Prozedur-parameter ausgegeben.

Die Parameterversorgung kann auch direkt beim Aufruf über Schlüsselworte erfolgen:

TSN=*tsn* Es werden nur die vom Task mit TSN *tsn* eröffneten Dateien berücksichtigt. Die Prozedur beendet sich automatisch, falls die angegebene TSN im System nicht mehr existiert.

FILE=*datei* Nur die IO's für die Datei *datei* berücksichtigt.

VOLUME=*vsn* Es werden nur Dateien berücksichtigt, die auf der Platte *vsn* mindestens einen Extent besitzen und eröffnet sind.

IO=*n* Die Ausgabe beschränkt sich auf alle Dateien, auf die im Meßintervall mehr als *n* IO's durchgeführt wurden.

SUMIO=TSN|SUM Die Ausgabe der IO's für die Dateien erfolgt einzeln für jede TSN (Standard) oder als Summe der IO's über alle TSN's, die auf diese Datei zugegriffen haben (SUM).

INT=*sec* Dauer des Meßintervalls in Sekunden (Standard: 30 sec).

JOBVAR=*jv-name* Name einer Jobvariablen, die zum Beenden der Messung verwendet werden kann. Die Jobvariable muß in Spalte 1 bis 3 auf den Wert 'END' gesetzt werden.

WFILE=*wfile* Die Informationen werden zusätzlich in die Datei *wfile* geschrieben.

Beispiel für eine FILEIO-Auswertung:

```
******************************************************************************
* F I L E - I O 'S        DATUM: 20.07.95    HOST: HOST11
******************************************************************************
* P A R A M E T E R  :    TSN       = *ALL
*                         FILENAME  = *ALL
*                         VOLUME    = *ALL
*                         DELTA-IO  = >0
*                         INTERVALL = 30
*                         SUMIO     = TSN
*-----------------------------------------------------------------------------
* TIME    FILENAME                                    D-IO  REA  WRI  TSN OPN
*-----------------------------------------------------------------------------
09:06:51  :CFS:$TT08.CFS.PLAM1                          18   18    0 0PVS I
09:06:51  :O:$TSOS.SYSLNK.ASSEMBH.011.LG                13   13    0 0PVV EX
09:06:51  :O:$TSOS.SYSSDF.ASSEMBH.011.USER               3    3    0 0PVV EX
09:06:51  :O:$TSOS.ASSEMBH                               3    3    0 0PVV EX
09:06:51  :CFS:$TT08.CFS.E.0PVS.090642                   3    2    1 0PVS
09:06:51  :CFS:$TT08.S.IN.CFS.PLAM1.ASS.0PVV.134253      2    2    0 0PVV EX
09:06:51  :CFS:$TT08.CFS.E.0PVS.090642                   1    1    0 0PVV EX
09:07:28  :CFS:$TT08.CFS.PLAM1                         128  128    0 0PVQ I
09:07:28  :CFS:$TT08.CFS.PLAM1                          64   64    0 0PVV
09:07:28  :MAC:$TSOS.MACROLIB.V10                       30   30    0 0PVV
09:07:28  :CFS:$TT08.S.214.0PVV.CIFLIB                  15    2   13 0PVV OI
09:07:28  :CFS:$TT08.S.214.0PVQ.ZS.WRTRD.0112            9    5    4 0PVQ
09:07:28  :O:$TSOS.SYSSDF.ASSEMBH.011.USER               3    3    0 0PVV EX
09:08:03  :MAC:$TSOS.MACROLIB.V10                       78   78    0 0PVV
09:08:03  :CFS:$TT08.S.214.0PVV.CIFLIB                  25    0   25 0PVV OI
09:08:03  :CFS:$TT08.CFS.PLAM1                          24   24    0 0PVV
09:08:37  :CFS:$TT08.CFS.PLAM1                          50   50    0 0PVV
09:08:37  :CFS:$TT08.S.214.0PVV.CIFLIB                  38    0   38 0PVV OI
09:08:37  :MAC:$TSOS.MACROLIB.V10                       17   17    0 0PVV
09:09:09  :CFS:$TT08.CFS.PLAM1                          46   46    0 0PVV
09:09:09  :CFS:$TT08.S.214.0PVV.CIFLIB                  34    0   34 0PVV OI
09:09:09  :MAC:$TSOS.MACROLIB.V10                       22   22    0 0PVV
09:09:43  :CFS:$TT08.S.214.0PVV.CIFLIB                  61    0   61 0PVV OI
09:09:43  :MAC:$TSOS.MACROLIB.V10                       12   12    0 0PVV
09:09:43  :CFS:$TT08.CFS.PLAM1                           7    7    0 0PVV
09:10:17  :CFS:$TT08.S.214.0PVV.CIFLIB                  58    0   58 0PVV OI
09:10:17  :CFS:$TT08.CFS.PLAM1                           8    0    8 0PVV
09:10:51  NO ENTRIES FOUND
09:11:22  :CFS:$TT08.S.214.0PVV.CIFLIB                 288  272   16 0PVV OI
09:11:22  :CFS:$TT08.CFS.PLAM1                          97   11   86 0PVV
09:11:56  :CFS:$TT08.CFS.MDCCRDX.SRC                   175    0  175 0PVV O
09:12:30  :CFS:$TT08.CFS.MDCCRDX.SRC                   377    0  377 0PVV O
09:13:04  :CFS:$TT08.CFS.MDCCRDX.SRC                   429    0  429 0PVV O
09:13:35  :CFS:$TT08.CFS.MDCCRDX.SRC                   428    0  428 0PVV O
09:14:09  :CFS:$TT08.CFS.MDCCRDX.SRC                   353    0  353 0PVV O
09:14:43  :CFS:$TT08.CFS.MDCCRDX.SRC                   255    0  255 0PVV O
09:15:17  :CFS:$TT08.CFS.MDCCRDX.SRC                   916  916    0 0PVV I
09:15:17  :CFS:$TT08.CFS.MDCCRDX.SRC.FLM               221    0  221 0PVV O
09:15:17  :O:$FLAM.FLAM                                 20   20    0 0PVV EX
09:15:17  :CFS:$TT08.CFS.PLAM1                           2    0    2 0PVV
09:15:17  :CFS:$TT08.S.IN.CFS.PLAM1.ASS.0PVV.134253      1    1    0 0PVV EX
09:15:17  :CFS:$TT08.S.IN.CFS.LIB.XXX.DO.FLAMK.0PVV.42   1    1    0 0PVV EX
09:15:52  NO ENTRIES FOUND
09:16:23  NO ENTRIES FOUND
```

/DO CFS.USERLIB.BEISPIEL(FILESTAT),...

Die BS2000-DO Prozedur vermißt eine einzelne TSN, eine Task mit einem vorgegebenen Jobnamen oder alle Tasks einer Benutzerkennung. Die Prozedur ermittelt zyklisch die Anzahl der IO's, die im vorgegebenen Meßintervall für jede bzw. ausgewählte geöffnete Dateien angefallen sind. Darüber hinaus werden die CPU-Verbrauchswerte und weitere Informationen der betroffenen Tasks ausgegeben. Die ermittelten Informationen werden in eine Datei geschrieben. Bei Überwachung eines bestimmten Jobnamens oder einer Benutzerkennung wartet die Prozedur mit der Messung, bis der erste zu vermessende Task gestartet wurde. Die Prozedur beendet sich automatisch, sobald keine zu vermessende Task mehr existiert. Die Messung kann auch vorzeitig beendet werden, indem eine vorgegebene Jobvariable vom Benutzer auf 'END' gesetzt wird.

Aufruf:

/DO $*user-id*.CFS.USERLIB.BEISPIEL(FILESTAT), ...

JOBNAME=<u>*NONE</u> | *jobnam*

Mit diesem Parameter kann ein Task mit einem vorgegebenen Jobnamen (z.B. TEST01) überwacht werden. Falls noch kein Task mit diesem Jobnamen existiert, wartet die Prozedur mit der Messung, bis der erste Task mit diesem Namen gestartet ist. Die Messung wird automatisch beendet, sobald der Task mit dem vorgegebenen Jobnamen auf LOGOFF läuft.

USERID=<u>*NONE</u> | *userid*

Mit diesem Parameter werden alle Tasks vermessen, die unter einer vorgegebenen Benutzerkennung angeloggt sind (z.B. TT08). Falls noch kein Task unter dieser Kennung existiert, wartet die Prozedur mit der Messung, bis der erste Task unter der Benutzerkennung gestartet ist. Die Messung wird automatisch beendet, sobald der letzte Task unter der Benutzerkennung auf LOGOFF läuft.

TSN=<u>*NONE</u> | *tsn*

Mit diesem Parameter kann ein laufender Task mit einer vorgegebenen TSN vermessen werden. Die Messung wird automatisch beendet, sobald der Task mit der vorgegebenen Jobnamen auf LOGOFF läuft.

Nur einer der oben beschriebenen drei Parameter darf bein Aufruf der Prozedur angegeben werden. Die anderen beiden müssen den Standardwert *NONE besitzen.

FILENAME=<u>*ALL</u> | *such*

Es werden nur die IO's von Dateien ausgegeben, deren Namen der in *such* spezifizierten CFS-Suchbedingung genügen.
Standard: *ALL. Es erfolgt keine Beschränkung bezüglich der Dateinamen.

INT=*sec* Das Meßintervall wird auf *sec* Sekunden gesetzt (Standard: 30 sec).

CFSPHASE=$CFS.CFS

Benutzerkennung und Dateiname der Programmphase von CFS.

JOBVAR=$CFS.CFS.JV1

Name einer Jobvariablen, die zum vorzeitigen Beenden der Messung verwendet werden kann. Die Jobvariable muß in Spalte 1 bis 3 auf den Wert 'END' gesetzt werden.

WFILE=FILESTAT.PROT

Die Ergebnisse der Messung werden in die angegebene Datei geschrieben.

Beispiel für eine FILESTAT-Auswertung:

```
---------------------------------------------------
I O  -  S T A T I S T I K   VOM   20.07.95  14:31:22
---------------------------------------------------
          USER-ID      : TT08
          INTERVALL    : 60 SEK
          FILES        : *ALL
----------------------------------------------------------------------------------------
TIME         REA  WRI FILENAME                                       D-IO   TSN  OPN
14:32:30       1    0 :CFS:$TT08.CFS.E.0PVS.143126                      1   0PVS
14:32:30       1    0 :CFS:$TT08.CFS.E.0PVS.143126                      1   0PVZ EX
14:32:30       2    0 :CFS:$TT08.S.IN.CFS.PLAM1.ASS.0PVZ.826748         2   0PVZ EX
14:32:30      85    0 :CFS:$TT08.CFS.PLAM1                             85   0PVZ
14:32:31     102    0 :MAC:$TSOS.MACROLIB.V10                         102   0PVZ
14:32:31       6    0 :O:$TSOS.SYSSDF.ASSEMBH.011.USER                   6   0PVZ EX
14:32:31       3    0 :O:$TSOS.ASSEMBH                                   3   0PVZ EX
14:32:31       2   38 :CFS:$TT08.S.214.0PVZ.CIFLIB                      40   0PVZ OI

TIME                TSN  USER-ID  JOBNAME                D-CPU     D-IO D-PAGING  D-OTHER
14:32:31            0PVQ TT08     HILBERT  240 3 DIA      0.24        0        0        0
14:32:32            0PVR TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:32:32            0PVS TT08     HILBERT  240 3 DIA      2.05        8        0        0
14:32:32            0PVZ TT08     HILBERT  255 2 BAT     45.84      259        0        0
14:32:32  SUM OF ALL DELTA IN INTERVALL:                 48.13      267        0        0
----------------------------------------------------------------------------------------
TIME         REA  WRI FILENAME                                       D-IO   TSN  OPN
14:33:33     101    0 :CFS:$TT08.CFS.PLAM1                            101   0PVZ
14:33:33      44    0 :MAC:$TSOS.MACROLIB.V10                          44   0PVZ
14:33:33       0   72 :CFS:$TT08.S.214.0PVZ.CIFLIB                      72   0PVZ OI

TIME                TSN  USER-ID  JOBNAME                D-CPU     D-IO D-PAGING  D-OTHER
14:33:34            0PVQ TT08     HILBERT  240 3 DIA      0.13        0        0        0
14:33:34            0PVR TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:33:34            0PVS TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:33:34            0PVZ TT08     HILBERT  255 2 BAT     52.31      240        0        0
14:33:34  SUM OF ALL DELTA IN INTERVALL:                 52.44      240        0        0
----------------------------------------------------------------------------------------
TIME         REA  WRI FILENAME                                       D-IO   TSN  OPN
14:34:37       7    7 :CFS:$TT08.CFS.PLAM1                             14   0PVZ
14:34:38      15    0 :MAC:$TSOS.MACROLIB.V10                          15   0PVZ
14:34:38       0  124 :CFS:$TT08.S.214.0PVZ.CIFLIB                     124   0PVZ OI

TIME                TSN  USER-ID  JOBNAME                D-CPU     D-IO D-PAGING  D-OTHER
14:34:38            0PVQ TT08     HILBERT  240 3 DIA      0.13        0        0        0
14:34:38            0PVR TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:34:38            0PVS TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:34:39            0PVZ TT08     HILBERT  255 2 BAT     53.79      193        1        0
14:34:39  SUM OF ALL DELTA IN INTERVALL:                 53.92      193        1        0
----------------------------------------------------------------------------------------
TIME         REA  WRI FILENAME                                       D-IO   TSN  OPN
14:35:40      11   89 :CFS:$TT08.CFS.PLAM1                            100   0PVZ
14:35:40     258   16 :CFS:$TT08.S.214.0PVZ.CIFLIB                     274   0PVZ OI

TIME                TSN  USER-ID  JOBNAME                D-CPU     D-IO D-PAGING  D-OTHER
14:35:41            0PVQ TT08     HILBERT  240 3 DIA      0.14        0        0        0
14:35:41            0PVR TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:35:41            0PVS TT08     HILBERT  240 3 DIA      0.01        0        0        0
14:35:41            0PVZ TT08     HILBERT  255 2 BAT     46.50      397        1        0
14:35:41  SUM OF ALL DELTA IN INTERVALL:                 46.65      397        1        0
----------------------------------------------------------------------------------------
TIME         REA  WRI FILENAME                                       D-IO   TSN  OPN
14:36:45       0  535 :CFS:$TT08.CFS.MDCCRDX.SRC                       535   0PVZ O

TIME                TSN  USER-ID  JOBNAME                D-CPU     D-IO D-PAGING  D-OTHER
14:36:45            0PVQ TT08     HILBERT  240 3 DIA      0.13        0        0        0
14:36:45            0PVR TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:36:46            0PVS TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:36:46            0PVZ TT08     HILBERT  255 2 BAT     57.63      572        0        0
14:36:46  SUM OF ALL DELTA IN INTERVALL:                 57.76      572        0        0
----------------------------------------------------------------------------------------
TIME         REA  WRI FILENAME                                       D-IO   TSN  OPN
14:37:47       0  853 :CFS:$TT08.CFS.MDCCRDX.SRC                       853   0PVZ O
```

```
TIME              TSN  USER-ID  JOBNAME                D-CPU     D-IO D-PAGING  D-OTHER
14:37:48          0PVQ TT08     HILBERT  240 3 DIA      0.14        0        0        0
14:37:48          0PVR TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:37:48          0PVS TT08     HILBERT  240 3 DIA      0.01        0        0        0
14:37:48          0PVZ TT08     HILBERT  255 2 BAT     57.42      899        0        0
14:37:48  SUM OF ALL DELTA IN INTERVALL:               57.57      899        0        0
-----------------------------------------------------------------------------------------
TIME        REA  WRI FILENAME                                    D-IO   TSN  OPN
14:38:52      0  622 :CFS:$TT08.CFS.MDCCRDX.SRC                   622   0PVZ O

TIME              TSN  USER-ID  JOBNAME                D-CPU     D-IO D-PAGING  D-OTHER
14:38:52          0PVQ TT08     HILBERT  240 3 DIA      0.13        0        0        0
14:38:52          0PVR TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:38:52          0PVS TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:38:53          0PVZ TT08     HILBERT  255 2 BAT     57.68      657        0        0
14:38:53  SUM OF ALL DELTA IN INTERVALL:               57.81      657        0        0
-----------------------------------------------------------------------------------------
TIME        REA  WRI FILENAME                                    D-IO   TSN  OPN
14:39:54      1    0 :CFS:$TT08.S.IN.CFS.PLAM1.ASS.0PVZ.826748      1   0PVZ EX
14:39:54      0    2 :CFS:$TT08.CFS.PLAM1                           2   0PVZ
14:39:54      1    0 :CFS:$TT08.S.IN.CFS.LIB.XXX.DO.FLAMK.0PVZ.08   1   0PVZ EX

TIME              TSN  USER-ID  JOBNAME                D-CPU     D-IO D-PAGING  D-OTHER
14:39:55          0PVQ TT08     HILBERT  240 3 DIA      0.13        0        0        0
14:39:55          0PVR TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:39:55          0PVS TT08     HILBERT  240 3 DIA      0.00        0        0        0
14:39:55          0PVZ TT08     HILBERT  255 2 BAT     33.81        4        0        0
14:39:55  SUM OF ALL DELTA IN INTERVALL:               33.94        4        0        0
-----------------------------------------------------------------------------------------
TIME        REA  WRI FILENAME                                    D-IO   TSN  OPN
14:40:56      2    0 :CFS:$TT08.CFS.JR.HILBERT.SYSOUT.0PVS.143126   2   0PVS

                                                         CPU       IO   PAGING    OTHER
14:40:56  TOTAL :                                     408.22     3229        2        0
14:40:56  ENDE DER MESSUNG
```

DO SPACE

Die CFS-Prozedur ermittelt für die in einer Reihe von Variablen hinterlegten Pubsets, Privatplatten oder Pools von Pubsets/Privatplatten, den aktuellen Belegungszustand und gibt die Informationen am Bildschirm aus. Für die einzelnen Pubsets, Privatplatten oder Pools können individuelle Grenzwerte vorgegeben werden. Falls diese Grenzwerte überschritten sind, werden die entsprechenden Werte bei der Ausgabe hell dargestellt.

Vor dem Start der Prozedur müssen die Pubsets, Privatplatten und Pools in der Prozedur angepaßt werden.

Die einzelnen Pubsets, Privatplatten und Platten-Pools sind innerhalb der Prozedur in Variablen &VOL1, &VOL2, &VOL3, ..., &VOL*n* fortlaufend zu hinterlegen. Der Inhalt der Variablen &VOL1, ... ,&VOL*n* wird folgendermaßen interpretiert:

*&VAR*n*='PVSI_VOLUME_POOLNAME_PRO%_PAMPAGES_POOLRELA'

PVSI	Catid eines Public-Volume-Sets
VOLUME	VSN einer Privatplatte
POOLNAME	Poolname, der über POOLRELA zusammengefassten Platten
PRO%	Grenzwert für die Plattenbelegung in Prozent
PAMPAGES	Grenzwert für die Anzahl freier PAM-Seiten
POOLRELA	Poolname für die Zuordnung zu einem Plattenpool

Aufruf:

DO SPACE [, PUB|PRIV|POOL] [, FILE=*datei*]

Ohne Angabe von Parametern werden für alle in der Prozedur hinterlegten Pubsets, Privatplatten und Plattenpools die aktuellen Belegungszustände am Bildschirm angezeigt.

PUB	Es werden nur die Informationen für die definierten Pubsets angezeigt.
PRIV	Es werden nur die Informationen für die definierten Privatplatten angezeigt.
POOL	Es werden nur die Informationen für die definierten Plattenpools angezeigt.
FILE=*datei*	Die angeforderten Informationen werden in eine SAM-Datei *datei* geschrieben. Die Bildschirmausgabe wird unterdrückt.

/E CFS.USERLIB.BEISPIEL(ADDRESS-SPACE)

Die Enter-Prozedur bietet die Möglichkeit, den noch freien System-Adreßraum zyklisch zu überwachen. Dies ist besonders dann von Nutzen, wenn es sporadisch zu Engpässen kommt. Über Prozedurvariablen kann eine Meßwertedatei angegeben werden, in die jeweils die aktuell ermittelten Werte ausgegeben werden. Falls der vorgegebene Grenzwert unterschritten ist, wird an der Konsole eine Meldung ausgegeben und zusätzlich eine Liste der User- und System-Tasks (je 10 Tasks mit CL3, CL4, CL5, CL6) zum Zeitpunkt des Engpasses ausgegeben.

Die Prozedur kann über /INTR *tsn* von der Konsole aus beendet werden.

Vor Start der Prozedur sollte der Name der Meßdatei, der Grenzwert für den Systemadreßraum und die Intervallzeit an die individuelle Umgebung angepaßt werden:

wfile	Name der Meßdatei, in der zyklisch die ermittelten Informationen abgelegt werden (Standard: CFS.UEB.ADDRSP).
lim	Grenzwert für die Anzahl freier 4K-Seiten im Systemadreßraum (Standard: 500).
int	Prüfintervall in Sekunden (Standard: 600).

DO MOD-MAXCPU

Die CFS-Prozedur ermöglicht es, für einen laufenden Task, nachträglich die maximale CPU-Zeit hochzusetzen bzw. das NTL Attribut zu vergeben.
Die Prozedur verzweigt in die Komponente TAS, wählt die entsprechende TSN aus und modifiziert über die symbolische Anzeige von Kontrollblöcken (%TCB,%JTBP) die notwendigen Felder im TCB (Task Control Block) und im JTBP (Job to be processed Block).

Aufruf:

DO MOD-MAXCPU [, *tsn* [, *time*]

Ohne Angabe von Parametern wird die TSN und die zu vergebende CPU-Zeit über Prompting angefordert.

tsn TSN des zu modifizierenden Tasks

time neu zu vergebende CPU-Zeit bzw. NTL (No Time Limit)

Die Prozedur MOD-MAXCPU sollte nicht mehr benutzt werden, das die entsprechende Funktion durch den TAS Action-Code T realisiert ist.

/E CFS.USERLIB.BEISPIEL(CAN-CONMAX)

Die Enter-Prozedur läuft unter der Kennung TSOS und überwacht die Typ 3 (Dialog) Tasks aller Benutzerkennungen außer TSOS bezüglich der seit Logon vergangenen Zeit (Connect Zeit).

Die Prozedur cancelt alle interaktiven Tasks, deren Logon-Zeitpunkt mehr als *ccc* Minuten zurück liegt. Über einen zweiten Parameter *www* (<= *ccc*) kann die Anzahl der Minuten eingestellt werden, ab der Tasks per Message Kommando vor einem möglichen Cancel gewarnt werden.

Am Anfang der Prozedurdatei sind die vom Benutzer veränderbaren Parameter aufgeführt.

&CONMAX=720	Maximale Connect-Zeit in Minuten. Alle Dialogtasks, die länger connectiert sind, werden von der Prozedur gecancelt.
&CONWARN=600	Warnzeit in Minuten. Alle Dialogtasks, die länger connectiert sind, werden in regelmäßigen Abständen (siehe &WAIT) von der Prozedur per Message-Kommando auf ein bevorstehendes Canceln hingewiesen.
&WAIT=600	Wartezeit in Sekunden. Um den durch die Prozedur verursachten Systemoverhead gering zu halten, wird zwischen den einzelnen Durchläufen der Prozedur ein Wartezyklus von &WAIT Sekunden zwischengeschaltet. Die Wartezeit bewirkt außerdem, daß Message-Kommandos an Tasks nur in diesen Intervallen abgesendet werden.
&USID=-TSOS	Benutzerkennungen, die von der Überwachung auszuschließen sind. Mehrere auszuschließende Benutzerkennungen können angegeben werden in der Form: -*user1*+-*user2*+- ... Es ist auch möglich, daß nur die Tasks von bestimmten angegebenen Benutzerkennungen überwacht werden sollen. Hierzu müssen die Kennungen angegeben werden in der Form: *user1*,*user2*, ...
&JOBT=3	Es werden nur Tasks des angegebenen Jobtypes überwacht. Jobtype 3 = Dialogtasks.
	In der Prozedur ist außerdem die User-Id und das Passwort für den Zugang zu der $CONSOLE-Anwendung einzutragen.

Der Enter-Prozess für diese Prozedur kann nur per Cancel-Kommando beendet werden.

/E CFS.USERLIB.BEISPIEL(CAN-CPUMAX)

Die Enter-Prozedur läuft unter der Kennung TSOS und überwacht die Typ 3 (Dialog) Tasks aller Benutzerkennungen außer TSOS bezüglich des CPU-Verbrauchs.

Die Prozedur überwacht interaktive Tasks (&JOBT=3, &PRI=>239, &USID=-TSOS), die in einem Meßintervall von &INT Sekunden (&INT=10) mehr als &CPUMAX CPU-Sekunden (&CPUMAX=8) verbraucht haben. Falls der Parameter &WARNING auf 'Y' gesetzt ist (Standard), werden die Tasks mehrmals (&WARNCNT=3) per Message vor einem drohenden Cancel gewarnt. Danach bzw. sofort, falls &WARNING='N' werden diese Tasks gecancelt.

Am Anfang der Prozedurdatei sind die vom Benutzer veränderbaren Parameter aufgeführt.

&INT=10	Meßintervall in Sekunden. Im Meßintervall wird die verbrauchte CPU-Zeit der Tasks analysiert.
&CPUMAX=8	Schwellwert für CPU-Zeit in Sekunden. Es werden alle Tasks gemerkt, die im Meßintervall mehr als die angegebene CPU-Zeit verbraucht haben.
&USID=-TSOS	Benutzerkennungen, die von der Überwachung auszuschließen sind. Mehrere auszuschließende Benutzerkennungen können angegeben werden in der Form: -*user1*+-*user2*+- ... Es ist auch möglich, daß nur die Tasks von bestimmten angegebenen Benutzerkennungen überwacht werden sollen. Hierzu müssen die Kennungen angegeben werden in der Form: *user1*,*user2*, ...
&JOBT=3	Es werden nur Tasks des angegebenen Jobtypes überwacht. Jobtype 3 = Dialogtasks.
&PRI=>239	Es werden nur Tasks mit einer Priorität im angegebenen Bereich analysiert.
&WARNING='Y'	Alle Tasks, die die erlaubte CPU-Zeit im Intervall überschritten haben, werden per Message Kommando vor einem drohenden Cancel gewarnt. Bei &WARNING='N' wird sofort ohne Warnung gecancelt. In diesem Fall ist auch der folgende Parameter &WARNCNT ohne Bedeutung.
&WARNCNT=3	Falls die CPU-Fresser in 3 aufeinander folgenden Meßintervallen die zulässige CPU-Zeit jeweils überschritten haben erfolgt ein Cancel. Falls in einem Meßintervall die CPU-Zeit nicht überschritten wurde, wird der interne Zähler für diesen Task auf 0 zurückgesetzt.
&WAIT=60	Wartezeit in Sekunden zwischen den einzelnen Analysezyklen. Um den durch die Prozedur verursachten Systemoverhead gering zu halten, wird zwischen den einzelnen Durchläufen der Prozedur ein Wartezyklus von &WAIT Sekunden zwischengeschaltet. Die Wartezeit bewirkt außerdem, daß die Probenentnahme für die verbrauchte CPU-Zeit und die Warnung an die Tasks per Message-Kommando nur in diesen Intervallen erfolgt. In der Prozedur ist außerdem die User-Id und das Passwort für den Zugang zu der $CONSOLE-Anwendung einzutragen.

Der Enter-Prozess für diese Prozedur kann nur per Cancel-Kommando beendet werden.

/E CFS.USERLIB.BEISPIEL(CAN-IDLE)

Die Enter-Prozedur läuft unter der Kennung TSOS und überwacht die Typ 3 (Dialog) Tasks aller Benutzerkennungen bezüglich des Zeitpunkts des letzten CPU-Verbrauchs.

Die Prozedur cancelt alle interaktiven Tasks, die seit mehr als &IDLE Minuten keinen Zuwachs an CPU-Zeit verursacht haben.

Am Anfang der Prozedurdatei sind die vom Benutzer veränderbaren Parameter aufgeführt.

&IDLE=300 Idle-Zeit in Minuten. Alle Tasks, die seit mehr als 300 Minuten keinen CPU-Zuwachs verzeichnet haben, werden gecancelt.
Es ist zu beachten, daß bei &IDLE keine allzugroßen Werte angegeben werden sollten, da auch Idle-Tasks in größeren Zeitabständen minimale Anteile an CPU-Zeit aufnehmen.

&WAIT=60 Wartezeit in Sekunden zwischen den einzelnen Analysezyklen. Um den durch die Prozedur verursachten Systemoverhead gering zu halten, wird zwischen den einzelnen Durchläufen der Prozedur ein Wartezyklus von &WAIT Sekunden zwischengeschaltet.

Der Enter-Prozess für diese Prozedur kann nur per Cancel-Kommando beendet werden.

DO SECURE-HSMS

Die CFS-Prozedur wählt alle Dateien der eigenen Kennung aus, die sich auf HSMS Storage Level S2 befinden, die also ausgelagert sind. Es wird eine BS2000-DO-Prozedur mit einem SECURE-Kommando für alle durch den Benutzer mit X markierten Dateien erstellt. Dies hat zur Folge, daß die Dateien von HSMS wieder eingelagert werden.

Die Einlagerung mehrerer Dateien hat den Vorteil, daß alle markierten Dateien mit **einem** Kommando angesprochen werden, und somit auch nur **ein** HSMS-Task zur Einlagerung erzeugt wird.

Die erzeugte Prozedur wird über das Job-Report-System als Enterprozess mit Überwachung gestartet.

Administrationsschnittstellen

1. Zentrale Prüfroutine USRMOD

Die Prüfroutine von CFS (Modul USRMOD) wird nach jeder Maskeneingabe durchlaufen und prüft die Eingaben. Bestimmte CFS-Funktionen können damit einem engeren Kreis von besonders autorisierten Benutzern zur Verfügung gestellt werden. Die Prüfung, ob eine CFS-Funktion einem Benutzer zur Verfügung steht, erfolgt tabellengesteuert. Die vom Systemverwalter zu modifizierenden Tabellen stehen sich im Datenteil der Prüfroutine USRMOD. Als für den Betrieb von CFS Verantwortlicher sollten Sie das Assembler-Quellprogramm USRMOD (gespeichert in der CFS.S.LMSLIB) genau studieren.

Beschreibung der Prüftabellen

Für jedes Eingabefeld einer Maske gibt es eine entsprechende Prüftabelle in USRMOD:

Selektionsmaske

Feld Filename-Select	TAB01
Feld User-Id	TAB02
....	
Feld Variable Action	TAB0C

Maske der Dateienliste/CFS-Editor

CFS-Kommandos	TAB10
Action-CodesTAB11.	

Jede Benutzereingabe in einem Feld einer CFS-Maske wird mit den in den entsprechenden Tabellen enthaltenen Mustereingaben verglichen. Bei Übereinstimmung wird aufgrund der User-Id-Tabelle (siehe unten) entschieden, ob die Eingabe unter der aktuellen User-Id erlaubt ist oder nicht (Authority-Reject).

Die Prüftabellen sind wie folgt aufgebaut:

Im Tabellenkopf befindet sich ein Default-Indikator, der festlegt, was zu geschehen hat, wenn die aktuelle Eingabe in der Prüftabelle für das Maskenfeld nicht enthalten ist (Eingabe akzeptieren bzw. zurückweisen). Im Anschluß an den Tabellenkopf folgen beliebig viele Tabellenelemente, die den zu prüfenden Benutzereingaben zugeordnet sind.

Struktur eines Tabellenelements

`DC AL1(len)`	Gesamtlänge des nachfolgenden Tabellenelements.
`DC C'ind'`	Indikator
C'␣'	der folgende String ist als Eingabe erlaubt.
C'-'	der folgende String ist als Eingabe nicht erlaubt und führt zu einem Authority-Reject.

C'R' — Es wird ein benutzerspezifischer Code zur Behandlung dieser Eingabe durchlaufen. Auf C'R' folgt unmittelbar die vier Byte lange Adresse der Unterroutine zur Behandlung der Eingabe.

`DC AL4(Sonderroutine)` — Nur falls *ind* = C'R': Adresse einer Unterroutine innerhalb von USERMOD, die bei Tätigung der unten angegebenen Eingabe an gesprungen wird.

`DC AL4(User-id Tab)` — optionale Adresse einer alternativen User-Id Tabelle für die aktuelle Eingabe (Funktion der User-ID-Tabelle: siehe unten).

`DC C'string'` — Muster der Benutzereingabe.

Das Zeichen '*' wird als Beliebigkeitszeichen gewertet und hat zur Folge, daß an den entsprechenden Stellen in *string* ein beliebiges Zeichen stehen kann. Z.B. steht C'OC*' für die Benutzereingaben OC0, OC1, ..., OC9.

User-ID-Tabellen

Neben den Prüftabellen für die Benutzereingaben existieren User-Id Tabellen. Eine **User-Id Tabelle** enthält eine Gruppe von Benutzerkennungen. Bei Übereinstimmung der Eingabe in einem Maskenfeld mit einem in der entsprechenden Prüftabelle (siehe oben) angegebenen Muster wird aufgrund der User-Id Tabelle entschieden, ob die Eingabe unter der aktuellen User-Id erlaubt ist oder nicht (--> Authority-Reject). Am Anfang jeder User-Id Tabelle steht ein Default-Indikator. Dieser bestimmt, wie zu verfahren ist, falls CFS unter einer nicht in der User-Id Tabelle aufgeführten Benutzerkennung aufgerufen wurde (Eingabe akzeptieren / zurückweisen).

Neben den in der Standard User-Id Tabelle zusammengefaßten Benutzerkennungen können beliebig viele weitere Gruppen von Benutzerkennungen in sog. **alternativen User-Id Tabellen** definiert werden. Für jede in einer Prüftabelle aufgeführte Mustereingabe kann im Prinzip eine eigene User-Id Tabelle verwendet werden.

Aktivierung der USRMOD-Prüfungen

Die Prüfroutine USRMOD ist standardmäßig und aus Performancegründen nicht aktiviert und wird von CFS daher nicht aufgerufen. Um USRMOD zu aktivieren, kann in der gebundenen Programmphase CFS im 6-ten PAM-Block an der Spalte 49 das Zeichen 'X' eingetragen werden.

Eine weitere Möglichkeit zum Aktivieren von USRMOD sowie zum Modifizieren einer Anzahl weiterer Installationsparameter von CFS besteht darin, das Assembler Quellprogramm CFSMAIN in der CFS.S. LMSLIB den eigenen Erfordernissen entsprechend anzupassen. I.A. sind lediglich die Kommentarsterne bei den MVI-/MVC-Anweisungen zu entfernen. Der modifizierte CFSMAIN ist anschließend durch den Prozeduraufruf /DO CFS.S.LMSLIB(ASSEMB) zu übersetzen. Zum Schluß muß mit Hilfe des Kommandos /DO CFS.S.LMSLIB(CFSLNK) eine neue Programmphase erzeugt werden.

Zusammenfassung

Der Systemverantwortliche kann mit Hilfe des Prüfmoduls USRMOD eine Reihe von Eingabemöglichkeiten zusammenstellen. Für jede dieser Eingaben kann eine Gruppe von User-Ids festgelegt werden, unter denen die Eingabe erlaubt bzw. nicht erlaubt ist (Authority-Reject).

Da alle Benutzereingaben über die Prüfroutine USRMOD laufen, bevor sie von CFS interpretiert und ausgeführt werden, können Eingaben des Benutzers im USRMOD auch abgeändert werden.

2. Datenschutz auf Benutzerebene

CFS trägt den Erfordernissen des Datenschutzes durch die im folgenden beschriebenen Einrichtungen Rechnung.

Eröffnen einer Connection: Bei Nachforderung des **Logon-Passworts** durch das BS2000 (PLEASE ENTER PASSWORD) wird dieses nicht in das Kommandogedächtnis der Connection eingetragen.

Bei Eingabe eines vollständigen LOGON-Kommandos inklusive Passwort nach OC*n* und 'PLEASE LOGON' wird das Passwort im Kommandogedächtnis der Connection mit C'SSS' überschrieben.

Einzelne Einträge des **Kommandogedächtnisses** (z.B. PASSWORD-Kommandos) können vom Benutzer explizit aus dem Gedächtnis gelöscht werden. Dies geschieht durch Betätigen der K3-Taste beim einzelschrittweisen Zurückgehen im Gedächtnis.

Connections: **LOG-Dateien**, die aufgrund der Eingabe LOG *datei* von CFS angelegt werden, sind grundsätzlich **nicht shareable**. Beim Logging in Nebenprozessen unter einer anderen Kennung oder auf einem anderen Host werden die LOG-Dateien stets unter der eigenen Benutzerkennung, d.h. dort wo CFS geladen wurde, angelegt.

LOCK: Sperren des Bildschirms bei kurzzeitiger Abwesenheit des Benutzers.

LOCK [*passw*] Der Bildschirm, an dem CFS geladen ist, wird gesperrt, bis der Benutzer im dunkel gesteuerten Feld das vereinbarte Lock-Passwort eingegeben hat. Der Benutzer kann sich von seinem Bildschirm entfernen, ohne daß eine andere Person in der Lage wäre, mit dem geladenen CFS weiter zu arbeiten, Connections zu benutzen oder im aktuellen Kommandogedächtnis zu blättern.
Wurde *passw* im LOCK-Kommando weggelassen, so gilt das zuletzt über SPL oder LOCK [*passw*] definierte Passwort.

SPL [*passw*] Set Password for Lock.

Es wird ein Passwort für ein später folgendes LOCK-Kommando definiert. Wird *passw* im SPL-Kommando weggelassen, so wird das Passwort in einem dunkel gesteuerten Feld angefordert.

3. Datenschutzmaßnahmen mittels USRMOD/CFSMAIN

CFS bietet den Systemverantwortlichen an verschiedenen Stellen Hilfen, die es ihm gestatten, das Verhalten der Benutzer unter CFS zu kontrollieren, zu protokollieren und in bestimmten Bereichen auch einzuschränken.

Eine Instanz zur Kontrolle der Benutzereingaben ist der Prüfmodul USRMOD. Falls die i.f. beschriebenen Einrichtungen für Ihr Rechenzentrum gewünscht werden, so sind diese im ausgelieferten Quellcode des Prüfmoduls USRMOD zu aktivieren. USRMOD muß anschließend neu assembliert und in die CFSLIB gebracht werden. Außerdem ist in der Programmphase CFS mit Hilfe eines Indikators (siehe auch Quellprogramm von CFSMAIN in der CFS.S.LMSLIB) der Ansprung des USRMOD durch CFS zu aktivieren.

Modify von PAM-Dateien nur gegen Eingabe eines Passworts

Falls der Benutzer unter einer Kennung verschieden von TSOS mit CFS arbeitet und eine im Display angezeigte PAM-Datei modifizieren möchte (Modify-Kommando), so muß er zuvor in einem dunkel gesteuerten Eingabefeld ein CFS-internes, vom CFS-Administrator änderbares Berechtigungspasswort eingeben. Dieses CFS-interne Passwort ist an der entsprechenden Stelle im Quellprogramm des Moduls USRMOD zu ersehen. Nach zweimaliger Falscheingabe des Passworts wird die Funktion abgewiesen (Authority-Reject). Um die Routine R100 zur Passwortanforderung zu aktivieren, sind in TAB10 von USRMOD in allen Statements die Kommentarmarkierungen '*2*' zu entfernen. Die vorbereitete Routine R100 kann dahingehend abgeändert werden, daß das Passwort beim Modify von Dateien mit bestimmten, in einer Tabelle abgelegten Namen angefordert wird.

Selektion durch FSTAT $user-id. im Feld FILENAME-SELECT unterbinden

Falls der Benutzer nicht unter TSOS arbeitet und über FSTAT $user-id bzw. FSTAT :x:$user-id Dateien einer fremden Kennung selektieren möchte, so wird dieser Selektionswunsch von CFS abgewiesen. Um die Routine R300 zur Prüfung der Filename-Select Eingabe auf FSTAT und anschließendes $-Zeichen zu aktivieren, sind in TAB01 von USRMOD in allen Statements die Kommentarmarkierungen '*3*' zu entfernen.

Selektion von Dateien auf festgelegten fremden Kennungen über das Feld USER-ID unterbinden

Falls der Benutzer nicht unter TSOS arbeitet und über das Feld USER-ID Dateien aus "verbotenen" Kennungen (definiert in der Tabelle R400TAB) selektieren möchte, so wird dieser Selektionswunsch von CFS abgewiesen. Um die Routine R400 zur Prüfung der Eingaben im Feld USER-ID zu aktivieren, sind in TAB02 von USRMOD in allen Statements die Kommentarmarkierungen '*4*' zu entfernen.

Virtuelle Terminalnamen für Connections entsprechend den physikalischen Stationsnamen

Standardmäßig bildet CFS die virtuellen Stationsnamen für Connections nach folgender Regel:

CFS *n xxxx*.

n ist dabei die Nummer der im OC-Kommando angegebenen Connection (0 <= *n* <= 9) und *xxxx* die TSN des Prozeßes, in dem das Programm CFS aufgerufen wurde. Diese virtuellen Terminalnamen erlauben keinen direkten Rückschluß auf die physische Datensichtstation, an die der Nebenprozeß gebunden ist (= physische Datensichtstation, an der CFS aufgerufen wurde).
Um eine klare Beziehung zwischen den unter CFS eröffneten Nebenprozessen und den zugehörigen physischen Datensichtstationen herzustellen, ist eine Unterroutine (R200) im Modul USRMOD vorbereitet.

Hierbei wird davon ausgegangen, daß die Bezeichnung aller Datensichtstationen an irgendeiner Stelle des acht Byte langen Namens ein bzw. zwei redundante, d.h. stets gleichbleibende Zeichen aufweist. Z.B. könnten die Bezeichnungen der Datensichtstationen mit einem konstanten Prefix beginnen. Es folgt ein variabler Teil, mit dem die Datensichtstationen eine fortlaufende Nummerierung erhalten.

Wir nehmen an, daß die Stationsnamen aller Datensichtgeräte lauten: DSS*xxxxx*. *xxxxx* ist hierbei der wesentliche, nicht redundante Namensbestandteil, über den jede Datensichtstation eindeutig identifiziert wird.

Die Routine R200 im Modul USRMOD bildet die Namen der an einem bestimmten Datensichtgerät eröffneten Nebenprozesse (Connections) nach folgender Regel: Der ursprüngliche Stationsname DSS*xxxxx* wird zu #*n*S*xxxxx* abgeändert. Dies bedeutet, daß die ersten beiden redundanten Stellen des ursprünglichen Stationsnamens ('DS') durch die Zeichen '#*n*' ersetzt werden. *n* ist hierbei die Nummer der im OC-Kommando gewählten Connection.

Die Routine R200 tauscht auch einen vom Benutzer im OC-Kommando explizit angegebenen Stationsnamen gegen einen Namen der oben beschriebenen Bildung aus. Der Benutzer hat somit keine Chance, einen von ihm selbst gewählten virtuellen Terminalnamen für einen CFS-Nebenprozeß zu vergeben. Diese Art der zwangsweisen Namensvergabe hat nebenbei noch den Vorteil, daß in einer Connection *n*, CFS durchaus wieder geladen werden kann. In diesem neuen CFS der Stufe 1 kann mit OC*n* aber nicht wieder eine Connection *n* eröffnet werden, da dies eine doppelte Verwendung des gleichen Stationsnamens bedeuten würde.

Der Austausch der Stationsnamen wird standardmäßig nur bei Connections zu $DIALOG vorgenommen. Bei Connections zu UTM-Anwendungen kann der Benutzer nachwievor seine eigenen Terminalnamen entsprechend der KDC-Generierung angeben. Durch Entfernen eines Kommentarsterns in der Unterroutine USRMOD kann der Austausch der Terminalnamen auch für UTM-, $CONSOLE- und DCAM-Anwendungen forciert werden.

Die i.f. beschriebene Möglichkeit zur Begrenzung der Nebenprozesse unter CFS ist unabhängig vom Prüfmodul USRMOD.

Begrenzung der Anzahl der möglichen Nebenprozesse

Über eine Modifikation in der gebundenen Programmphase CFS (siehe auch Quellprogramm des Moduls CFSMAIN in der CFS.S.LMSLIB) kann auf einfache Weise die maximale Anzahl der unter CFS zu eröffnenden Nebenprozesse beschränkt werden.

Durch Eintragen einer Zahl n ($0 <= n <= 9$) in der 6. PAM-Seite der gebundenen Programmphase CFS in der Spalte 48 wird die maximale Anzahl der zu eröffnenden Connections auf n begrenzt. Für TSOS bestehen keine Einschränkungen.

Die Zahl n kann auch bei den Initialisierungsparametern im Modul CFSMAIN angegeben werden.

Beispiel: Durch Eintragen der Zahl 2 können maximal zwei Connections, z.B. 1 und 2 oder 3 und 9 eröffnet werden. Wie dieses Beispiel zeigt, ist die Anzahl der Connections, nicht aber die höchste Connection-Nummer beschränkt.

Abweisen von Connections der Stufe 2

Von einer Connection der Stufe 2 sprechen wir, wenn der Grundprozeß zu einer CFS-Connection ebenfalls wieder als Nebenprozeß eines übergeordneten Task abläuft.

Über eine Modifikation in der gebundenen Programmphase CFS (6. PAM-Seite, Spalte 40: C'N'; siehe auch Quellprogramm von CFSMAIN in der CFS.S.LMSLIB) kann auf einfache Weise verhindert werden, daß Benutzer Connections der Stufe 2 eröffnen.

Rückkehr aus einer Connection nur bei Eingabeaufforderung

Um **Data-Stagnation** Fehler im Zusammenhang mit eröffneten Connections zu vermeiden, kann festgelegt werden, daß der Benutzer eine Connection nur dann verlassen kann, wenn nachfolgend keine freilaufenden Ausgaben mehr zu erwarten sind. Dieser Zustand ist immer dann gegeben, wenn vom Benutzer eine, wie auch immer geartete Eingabe angefordert wird. Modifikation (6. PAM-Seite von CFS, Spalte 67): C'X'.

4. Überwachung der Open Connection-Anforderung mit Hilfe einer Terminaldatei

In der Benutzerkennung, in der CFS installiert ist, also z.B. unter $CFS kann der Systemverwalter eine zentrale Datei `CFSTERM` (Fcbtype=Sam, Recform=V, Share=Yes) einrichten. In dieser Datei wird festgelegt, unter welchen Benutzerkennungen bzw. an welchen Datensichtstationen Connections zu welchen Anwendungen, Host-Rechnern und Benutzerkennungen eröffnet werden dürfen.

In der CFSTERM können 3 Arten von Prüfsätzen angegeben werden.

- Gewöhnliche Prüfsätze
- Default-Sätze
- Common-Sätze.

Die **gewöhnlichen Prüfsätze** haben folgenden Aufbau:

```
XUUUUUUUUBBXSSSSSSSSXCXPPPPPPPPXTTTTTTTTXHHHHHHHHXU1U1U1U1U2U2U2U2U3U3...
```

Zur Spalten-Orientierung bei der Erstellung der TERMFILE kann als erstes ein Satz dieser Art in die Datei eingebracht werden. Dieser Skalierungssatz ist nur für Editierzwecke und wird von CFS ignoriert.

Die Bedeutung der einzelnen Datenfelder:

`UUUUUUUU` User-Id. Dieses Feld wird mit der User-Id verglichen, unter der CFS geladen wurde. Werden für U...U Spaces eingetragen, so ist die User-Id, unter der CFS geladen wurde, irrelevant (beim Vergleich wird stets ein Treffer gemeldet). Für jede Position innerhalb von U...U kann das Sonderzeichen '*' angegeben werden. '*' ist das Beliebigkeitszeichen und bedeutet, daß die entsprechende Stelle der User-Id für den Vergleich nicht herangezogen wird.

`X vor U...U` Auswahlindikator.

x = Space positive Auswahl. Es wird ein Treffer gemeldet, wenn U...U mit der User-Id übereinstimmt, oder wenn U...U auf Spaces oder '********' gesetzt ist.

x = '-' negative Auswahl. Es wird ein Treffer gemeldet, wenn U...U **nicht** mit der User-Id übereinstimmt, unter der CFS geladen wurde.

`BB` Sonderindikator für U...U

`BB = 'P '` Im Feld U...U steht der Name des Datenstationsrechners, an dem die Datensichtstation angeschlossen ist und nicht die User-Id, unter der CFS geladen wurde. Mit U...U und S...S (siehe unten) ist die Datensichtstation somit im Netz eindeutig identifiziert.
Der P-Indikator ist von Nutzen, falls mehrere Datenstationen innerhalb eines LAN's existieren, die alle den gleichen Stationsnamen besitzen und sich nur durch ihren Prozessornamen unterscheiden.

`SSSSSSSS` Stationsname. Dieses Feld wird mit dem Namen der Datensichtstation, an der CFS geladen wurde, verglichen.

`X vor S...S` Auswahlindikator.

x = Space positive Auswahl. Es wird ein Treffer gemeldet, wenn S...S mit der Station übereinstimmt, oder wenn S...S auf Spaces oder '********' gesetzt ist.

x = '-' negative Auswahl. Es wird ein Treffer gemeldet, wenn S...S **nicht** mit der Station übereinstimmt.

`C` Connection-Nummer. Dieses Feld wird mit der im OC-Kommando angegebenen Nummer der Connection verglichen (1 bei OC1, usw.).
Steht an der Stelle C ein Blank, so kann die Connection-Nummer beliebig sein.

`X vor C` Auswahlindikator.

x = Space Falls die im OC-Kommando angegebene Nummer mit der in C angegebenen Zahl übereinstimmt, werden die nachfolgenden Bedingungen geprüft.

x = '-' Falls die im OC-Kommando angegebene Nummer **nicht** mit der in C angegebenen Zahl übereinstimmt, werden die im gleichen Satz nachfolgenden Bedingungen geprüft.

x = '<' Falls die im OC-Kommando angegebene Nummer **kleiner** ist als die im Feld C angegebene Zahl, werden die nachfolgenden Bedingungen geprüft. Insbesondere: Folgt auf '<' als letztes das Zeichen '-', so sind keine Connections mit einer Nummer kleiner als der unter C angegebenen erlaubt. Beispiel siehe Seite A659 und folgende.

x = '>' Falls die im OC-Kommando angegebene Nummer **größer** ist als die im Feld C angegebene Zahl, werden die nachfolgenden Bedingungen geprüft. Insbesondere: Folgt auf '>' als letztes das Zeichen '-', so sind keine Connections mit einer Nummer größer als der unter C angegebenen erlaubt. Beispiel siehe Seite A659 und folgende.

`PPPPPPPP` Partner. Dieses Feld wird mit dem Namen der im Nebenprozeß eröffneten Anwendung verglichen.

Standardwert bei Weglassung des entsprechenden Parameters im OC-Kommando ist '$DIALOG'. Weitere übliche Partnernamen können z.B. sein $CONSOLE oder Namen von UTM-/SAP-Anwendungen.

`X vor P...P` Auswahlindikator.

x = Space Der im OC-Kommando angegebene Partner muß mit dem in P...P angegebenen Partner übereinstimmen.

x = '-' negative Auswahl: Sind im Feld `PPPPPPPP` Blanks eingetragen, so wird durch die Angabe '-' bestimmt, daß keine Connections eröffnet werden können (Connections zu beliebigen Partnern sind nicht erlaubt).

x = 'C' Conditional: Falls der Name der Anwendung mit dem in P...P angegebenen Namen übereinstimmt, so werden die im gleichen Satz nachfolgenden Bedingungen geprüft.

x = 'N' Negative conditional: Falls der Name der Anwendung **nicht** mit dem in P...P angegebenen Namen übereinstimmt, so werden die im gleichen Satz nachfolgenden Bedingungen geprüft.

`TTTTTTTT` Virtueller Terminalname. Dieses Feld wird mit dem von CFS generierten bzw. vom Benutzer angegebenen Stationsnamen für den Nebenprozeß verglichen.
Standardwert bei Weglassung des entspr. Parameters im OC-Kommando ist CFS*ntttt* (*n* = Nummer der Connection, *tttt* = TSN des Basisprozeßes).

`X vor T...T` Auswahlindikator.

x = Space Der im OC-Kommando angegebene Terminalname muß mit dem in T...T angegebenen Terminal-Namen übereinstimmen.

x = '-' negative Auswahl: Sind im Feld `TTTTTTTT` Blanks eingetragen, so wird durch die Angabe '-' bestimmt, daß keine Connections eröffnet werden können (Connections mit beliebigen Terminalnamen sind nicht erlaubt).

x = 'C' Conditional: Falls der virt. Terminalname mit dem in T...T angegebenen Namen übereinstimmt, so werden die nachfolgenden Bedingungen geprüft.

x = 'N' Negative conditional: Falls der virt. Terminalname **nicht** mit dem in T...T angegebenen Namen übereinstimmt, so werden die nachfolgenden Bedingungen geprüft.

`HHHHHHHH` Hostname. Dieses Feld wird mit dem Namen desjenigen Hostrechners verglichen, auf dem der zu eröffnende Nebenprozeß erzeugt werden soll. Als Platzhalter für den eigenen Host-Rechner kann die Zeichenfolge '`$SAME   `' angegeben werden.

`X` `vor H...H` Auswahlindikator.

x = Space Der im OC-Kommando angegebene Host-Name muß mit dem in H...H angegebenen Host-Namen übereinstimmen.

x = '-' negative Auswahl: Sind im Feld `HHHHHHHH` Blanks eingetragen, so wird durch die Angabe '-' bestimmt, daß keine Connections eröffnet werden können (Connections zu beliebigen Hostrechnern sind nicht erlaubt).

x = 'C' Conditional: Falls der Host-Name mit dem in H...H angegebenen Namen übereinstimmt, so werden die nachfolgenden Bedingungen geprüft.

x = 'N' Negative conditional: Falls der Host-Name nicht mit dem in H...H angegebenen Namen übereinstimmt, so werden die nachfolgenden Bedingungen geprüft.

`U1U1U1U1` User-Id des Nebenprozeßes. Dieses Feld wird mit der User-Id des zu eröffnenden Nebenprozeßes verglichen (nur bei Connections zu $DIALOG).
Als Platzhalter für die eigene Benutzerkennung kann die Zeichenfolge '`$SAME   `' angegeben werden. Bei Eröffnung einer Connection zu $CONSOLE bzw. zu einer UTM-/SAP-Anwendung wird der Inhalt des Feldes U1...U1 nicht bewertet.

`X` `vor U1...U1` Auswahlindikator.

x = Space Die Benutzerkennung, unter der der Anwender in der CFS-Connection arbeiten will, muß mit U1...U1 übereinstimmen.

x = '-' Die Benutzerkennung, unter der der Anwender arbeiten will, darf nicht U1...U1 sein.

`U2U2U2U2` zweite User-Id für Nebenprozeß.
In einem Prüfsatz können bis zu 25 Benutzerkennungen hintereinander angegeben werden. Der Auswahlindikator vor U1...U1 bezieht sich auf alle angegebenen User-IDs.

Hinweise: Für jede zu prüfende Bedingung ist ein eigener Satz in der CFS-Termfile vorzusehen. Eine Kombination mehrerer abzuweisender Bedingungen in einem Datensatz ist nicht möglich.

Bei Verbindungen zu $DIALOG wird die Connection bei negativem Prüfergebnis erst nach erfolgtem LOGON abgewiesen.

Default-Sätze

Wird für die aktuelle Umgebung kein gewöhnlicher Prüfsatz in der Termfile gefunden, so werden, falls vorhanden, die Default-Sätze ausgewertet. Default-Sätze beginnen im Feld `XUUUUUUUU` mit der Record-Id '`$DEFLT  `'.

Bei Default-Sätzen werden nur die Felder `XC` bis `XUn...Un` zur Prüfung herangezogen. Es können beliebig viele Default-Sätze in der Termfile vorkommen. Sie werden stets alle der Reihe nach durchgeprüft. Erfüllt der aktuelle Connectionwunsch mindestens eine der Default-Bedingungen **nicht**, so wird er abgewiesen.

Common-Sätze

Werden für die aktuelle Umgebung gewöhnliche Prüfsätze in der Termfile gefunden, so werden diese zuerst der Reihe nach geprüft. Falls Common-Sätze in der Termfile vorhanden sind, so werden diese anschließend geprüft. Common-Sätze werden nicht ausgewertet im Default-Fall. Common-Sätze beginnen im Feld `XUUUUUUUU` mit der Record-Id '`$COM    `'.

Bei Common-Sätzen werden nur die Felder `XC` bis `XUn...Un` zur Prüfung herangezogen. Es können beliebig viele Common-Sätze in der Termfile vorkommen.

Ablauf des Prüfvorgangs

Es werden anhand der User-Id, der Station, unter welcher CFS geladen wurde, sowie der im OC-Kommando angegebenen Connection-Nummer (= "Umgebung") alle Prüfsätze in der Termfile gesucht, deren Felder `XUUUUUUUU`, `XSSSSSSSS` und `XC` mit der aktuellen Umgebung übereinstimmen.

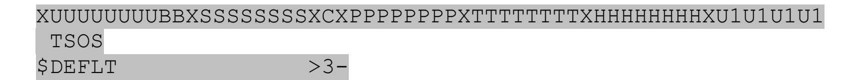
Falls der Auswahlindikator eines der Felder `P...P`, `T...T` oder `H...H` den Wert 'C' (= Conditional) besitzt, so wird auch dieses Feld zur Bestimmung der Umgebung herangezogen. Die nachfolgenden Felder in den Prüfsätzen der Termfile (`XP...P` bis `XUn...Un`) bestimmen, ob die Eröffnung der Connection erlaubt ist. Wenn dies nicht der Fall ist, so wird von CFS ein entsprechendes DC-Kommando generiert und die Verbindung wird abgebrochen. Bei $DIALOG geschieht dies aber erst nach dem vollständig (incl. Passwort) eingegebenen und ausgeführten Logon-Kommando.

Zu einer Umgebung (User-Id, Name der Datensichtstation, sowie Open-Connection-Nummer), bzw. zu einer teilqualifizierten Umgebung, können auch mehrere Prüfsätze in der Termfile existieren. In diesem Fall werden alle passenden Prüfsätze der Reihe nach ausgewertet. Nur wenn keine der Prüfungen ein negatives Ergebnis liefert, wird das Eröffnen der Connection erlaubt.

Die in der Termfile enthaltenen Datensätze müssen in keiner festgelegten Reihenfolge angegeben werden. Einzige Ausnahme: siehe das letzte der folgenden Beispiele.

Pro Datensatz kann in den Feldern XP...P bis XU1...U1 höchstens einmal der Negativindikator '-' angegeben werden. Eine Kombination von mehreren Negativbedingungen in einem Satz wird bei der Prüfung ignoriert.

Beispiele:



```
XUUUUUUUUBBXSSSSSSSSXCXPPPPPPPPXTTTTTTTTXHHHHHHHHXU1U1U1U1
 TSOS
$DEFLT              >3-
```

Wurde CFS unter einer Kennung ungleich TSOS geladen, so können keine Connections mit einer Nummer > 3 eröffnet werden (für alle Connections > 3 sind beliebige Partnernamen **nicht** erlaubt: `XPPPPPPPP` = '`-       `'). Wurde CFS unter TSOS geladen, so können ohne Einschränkung alle Connections von 0 bis 9 zu beliebigen Partnern eröffnet werden.

```
XUUUUUUUUBBXSSSSSSSSXCXPPPPPPPPXTTTTTTTTXHHHHHHHHXU1U1U1U1
 TEST                 -$DIALOG
```

Für die Kennung TEST sind keine Connections zu $DIALOG erlaubt.

```
XUUUUUUUUBBXSSSSSSSSXCXPPPPPPPPXTTTTTTTTXHHHHHHHHXU1U1U1U1U2U2U2U2
            DSS4711                     CVAR1     $SAME   ABC
            DSS4711                     CVAR2     $SAME   XYZ
```

Wurde CFS an der Datensichtstation DSS4711 geladen und wird per OC-Kommando ein Dialogprozeß zum Host VAR1 eröffnet (C vor VAR1: positiv conditional), so sind für das Logon-Kommando nur die eigene Benutzerkennung, sowie die zusätzliche Benutzerkennung ABC möglich. Bei Verbindungen zum VAR2 sind Nebenprozesse nur unter der eigenen, sowie unter der zusätzlichen Kennung XYZ möglich. Die Liste der zusätzlichen Benutzerkennungen kann auf bis zu 25 Einträge erweitert werden. Jeder Eintrag ist 8 Byte lang.

```
XUUUUUUUUBBXSSSSSSSSXCXPPPPPPPPXTTTTTTTTXHHHHHHHHXU1U1U1U1
            D4711
            D4712
            D4713
            D4714
$DEFLT                -$CONSOLE
```

Das Eröffnen von Connections zu $CONSOLE ist nur von Datensichtstationen D4711 bis D4714 möglich.

```
XUUUUUUUUBBXSSSSSSSSXCXPPPPPPPPXTTTTTTTTXHHHHHHHHXU1U1U1U1
                      C$CONSOLE CFSCON*
$DEFLT                -$CONSOLE
```

Falls eine Connection zu $CONSOLE eröffnet wird (C vor $CONSOLE: positiv conditional), so ist im OC-Kommando zwingend ein Terminalname von der Form CFSCONx (x: beliebiges Zeichen) anzugeben.

```
XUUUUUUUUBBXSSSSSSSSXCXPPPPPPPPXTTTTTTTTXHHHHHHHHXU1U1U1U1
-TSOS       CFS*****  -
```

Wurde CFS nicht unter TSOS geladen, so können Nebenprozesse ohne Einschränkung eröffnet werden. Wird in einem dieser Nebenprozesse jedoch wieder CFS aufgerufen und ein OC-Kommando für eine beliebige Anwendung abgegeben, so wird dieses abgewiesen, sofern als virtueller Stationsname der von CFS generierte Standardname CFS*nxxxx* verwendet wurde.

```
XUUUUUUUUBBXSSSSSSSSXCXPPPPPPPPXTTTTTTTTXHHHHHHHHXU1U1U1U1
$DEFLT              >3-
$DEFLT                -$CONSOLE
$DEFLT                                   $SAME
$DEFLT                                            $SAME   TEST
 TSOS               <7
 TSOS               >6-
 ABCD                                    $SAME
 ABCD               >6-
 A*******                                $SAME    $SAME
 A*******           >6-
 XXX                                     $SAME    $SAME   XYZ
 XXX                >6-
```

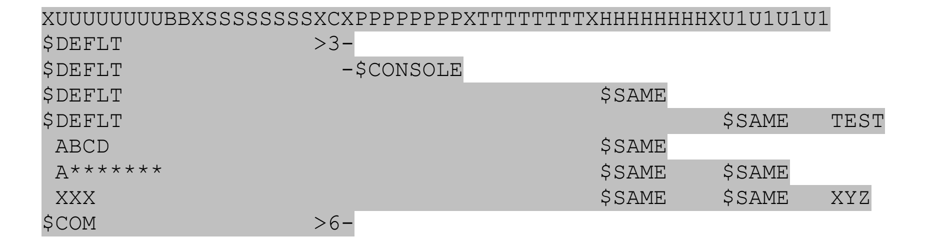
Für **alle Kennungen außer** TSOS, ABCD, beliebige Kennungen, die mit 'A' beginnen und XXX gilt:

- Es sind nur die Connections 0 bis 3 erlaubt. Es sind keine Connections zu $CONSOLE erlaubt.
- Es sind nur Connections im eigenen Host möglich.
- Es sind nur Dialogprozesse unter der jeweils eigenen, sowie unter der zusätzlichen Kennung TEST möglich.
- Für TSOS sind nur Connections 7 bis 9 gesperrt.
- Für die Kennung ABCD gilt das gleiche, jedoch sind Nebenprozesse nur am eigenen Host möglich.
- Für beliebige andere mit 'A' beginnende Kennungen sind Connections 7 bis 9 nicht erlaubt. Außerdem sind Nebenprozesse nur am eigenen Host und unter der eigenen Benutzerkennung möglich.
- Für die Kennung XXX gilt das gleiche, jedoch sind Nebenprozesse zusätzlich zur eigenen auch noch unter der Benutzerkennung XYZ erlaubt.

Hinweis:

Die **Reihenfolge** der Sätze in der Termfile ist für das Ergebnis der Prüfung nur insofern von Bedeutung, als die Sätze mit 'A*******' nach den Sätzen mit einer vollqualifizierten Benutzerkennung (z.B. 'ABCD␣␣␣␣') angegeben werden müssen. Die gleiche Regelung trifft auch für das Feld S...S zu.

Durch die Verwendung von Common-Sätzen ($COM) kann das obige Beispiel auch in einer verkürzten Form realisiert werden:



```
XUUUUUUUUBBXSSSSSSSSSXCXPPPPPPPPXTTTTTTTTXHHHHHHHHXU1U1U1U1
$DEFLT              >3-
$DEFLT                -$CONSOLE
$DEFLT                                   $SAME
$DEFLT                                            $SAME   TEST
 ABCD                                    $SAME
 A*******                                $SAME    $SAME
 XXX                                     $SAME    $SAME   XYZ
$COM                >6-
```

Beispiele für Termfile-Prüfungen finden Sie auch in der CFS.S.LMSLIB.

Hinweise zur Erstellung und Wartung der CFS-Termfile

- Um eine Termfile erstmalig zu erstellen, selektieren Sie mit Hilfe von CFS das Element TERMF0 aus der CFS.S.LMSLIB: SEL TERMF0,CFS.S.LMSLIB,CFSTERM . Der Skalierungssatz in der so erzeugten Datei CFSTERM dient als Orientierungshilfe für die von Ihnen neu einzutragenden Prüfsätze.
- Die Termfile muß shareable sein, so daß CFS von allen Benutzerkennungen darauf zugreifen kann. Falls im Modul CFSMAIN keine besonderen Eintragungen bezüglich des Namens der Termfile vorgenommen wurden, muß die Termfile unter der CFS-Kennung (z.B. unter $CFS) eingerichtet werden.

- Die Termfile kann empfindliche Daten, wie z.B. Benutzerkennungen, Namen von Datensichtstationen usw. enthalten. Diese Informationen sollten nicht im Klartext für jedermann zugänglich sein. Aus diesem Grunde sind die in der Termfile enthaltenen Daten zu verschlüsseln. Von CFS werden nur die verschlüsselten Informationen verarbeitet.

  Der Vorgang des Verschlüsselns geschieht, indem der CFS-Administrator unter TSOS die Termfile zuerst entsperrt (/VERIFY CFSTERM,REPAIR=ABS). Anschließend ist folgendes Kommando einzugeben: /EXEC (CONCH,$TSOS.CFSLIB). Auf die Frage nach der Input-Datei ist der Name der im Klartext vorliegenden Termfile anzugeben. Auf die Frage nach der Output-Datei ist der Name der zuvor entsperrten produktiven Termfile anzugeben. Die erfolgreiche Umsetzung wird anschließend bestätigt.

- Das entschlüsseln einer produktiven Termfile ist ebenfalls mit dem Modul CONCH möglich. Als Eingabedatei ist in diesem Fall die verschlüsselte Termfile und als Ausgabedatei die Datei im Klartext anzugeben. Das Verschlüsselungsverfahren ist umkehrbar.

- Aus Gründen der Datensicherheit sollte die Termfile vom CFS-Administrator auf ACCESS= READ gesetzt werden. In CFSMAIN kann außerdem ein READ-Passwort für die Termfile vereinbart werden. Vor jedem Zugriff auf die Termfile wird dieses Passwort von CFS abgesetzt und nach erfolgreichem Open sofort wieder freigegeben. Das ursprüngliche CAT-Kommando für die Termfile muß der Systemverwalter selbst vornehmen: /CAT CFSTERM,STATE=U,RDPASS=....

- Die Durchführung der in diesem Abschnitt beschriebenen OC-Prüfungen ist nur an das Vorhandensein der Termfile gebunden. Es sind weder Modifikationen im USRMOD vorzunehmen, noch sind irgendwelche Indikatoren im Modul CFSMAIN zu setzen. Es ist jedoch darauf zu achten, daß die Termfile den richtigen Dateinamen aufweist. Der benötigte Dateinamen kann ermittelt werden, indem nach Aufruf von CFS in der zweiten Maske mit Hilfe der K2-Taste in den Systemmodus verzweigt wird. Durch das AID-Kommando %D V'22A9'%XL30 wird der Name der benötigten Termfile angezeigt.

5. Überwachung der CFS-Programmaufrufe mit Hilfe einer Userdatei

Der Systemverwalter kann eine zentrale Datei $*user-id*.CFSUSER erstellen (FCBTYPE=SAM, RECFORM =V, *user-id* ist hierbei die Benutzerkennung, unter der CFS installiert ist). In dieser Datei wird festgelegt, unter welchen User-IDs bzw. an welchen Datensichtstationen ein /EXEC $CFS durch den Benutzer erlaubt bzw. nicht erlaubt ist. In der Datei CFSUSER kann eine beliebige Anzahl von Prüfsätzen angegeben werden.

Die **Prüfsätze** haben folgenden Aufbau:

`XUUUUUUUUBBXSSSSSSSS`

(Zur Spalten-Orientierung bei der Erstellung der TERMFILE kann als erstes ein Satz dieser Art in die Datei eingebracht werden. Dieser Skalierungssatz ist nur für Editierzwecke und wird von CFS ignoriert).

Die Bedeutung der einzelnen Datenfelder:

`UUUUUUUU` User-Id. Dieses Feld wird mit der User-Id verglichen, unter der der Benutzer CFS aufgerufen hat. Werden bei U...U Spaces eingetragen, so ist die User-Id, unter der CFS geladen wird, irrelevant. In diesem Fall ist nur die nachfolgende Spalte S...S (Datensichtstation) für die Prüfung der CFS-Berechtigung von Bedeutung.
Für jede Position innerhalb von U...U kann das Sonderzeichen '*' angegeben werden. '*' ist das Beliebigkeitszeichen und bedeutet, daß die entsprechende Stelle der User-Id für den Vergleich nicht herangezogen wird.

`X vor U...U` Auswahlindikator.

x = Space — positive Auswahl. Der Aufruf von CFS ist erlaubt, falls die User-ID, unter der sich der Benutzer angeloggt hat, mit der in U...U angegebenen Benutzerkennung übereinstimmt oder wenn U...U auf Spaces oder '********' gesetzt ist.

x = '-' — negative Auswahl. Der Aufruf von CFS wird zurückgewiesen, falls die User-ID, unter der sich der Benutzer angeloggt hat, mit der in U...U angegebenen Benutzerkennung übereinstimmt oder wenn U...U auf Spaces oder '********' gesetzt ist.

x = 'C' — Conditional: **Falls** die User-ID, unter der sich der Benutzer angeloggt hat, mit der in U...U angegebenen User-ID übereinstimmt, so wird der in S...S angegebene Name mit dem Namen der Datensichtstation verglichen, unter der CFS aufgerufen wird.

x = 'N' — Negative conditional: **Falls** die User-ID, unter der sich der Benutzer angeloggt hat, **nicht** mit der in U...U angegebenen User-ID übereinstimmt, so wird der in S...S angegebene Name mit dem Namen der Datensichtstation verglichen, unter der CFS aufgerufen wird.

`BB` konstant Spaces.

`SSSSSSSS` Stationsname. Dieses Feld wird mit dem Namen der Datensichtstation, an der CFS geladen wurde, verglichen.

Für S...S und den davorstehenden Auswahlindikator X gelten analog alle für U...U gemachten Bemerkungen.

Werden bei S...S Spaces eingetragen, so ist die Datensichtstation, an der CFS geladen wird, irrelevant. In diesem Fall ist nur die Spalte U...U (User-ID) für die Prüfung der CFS-Berechtigung von Bedeutung.
Für jede Position innerhalb von S...S kann das Sonderzeichen '*' angegeben werden. '*' ist das Beliebigkeitszeichen und bedeutet, daß die entsprechende Stelle im Stationsnamen für den Vergleich nicht herangezogen wird.

`X vor S...S` Auswahlindikator.

x = Space — positive Auswahl. Der Aufruf von CFS ist erlaubt, falls die Datensichtstation, unter der sich der Benutzer angeloggt hat, mit dem in S...S angegebenen Namen übereinstimmt oder wenn S...S auf Spaces oder '********' gesetzt ist.

x = '-' — negative Auswahl. Der Aufruf von CFS wird zurückgewiesen, falls die Datensichtstation, unter der sich der Benutzer angeloggt hat, mit dem in S...S angegebenen Namen übereinstimmt oder wenn S...S auf Spaces oder '********' gesetzt ist.

x = 'C' — Conditional: Falls die Datensichtstation des Benutzers mit der in S...S angegebenen Datensichtstation übereinstimmt, so wird die bei U...U eingetragene User-ID mit der Benutzerkennung verglichen, unter der sich der Benutzer angeloggt hat.

x = 'N' — Negative conditional: Falls die Datensichtstation des Benutzers nicht mit der in S...S angegebenen Datensichtstation übereinstimmt, so wird die bei U...U eingetragene User-ID mit der Benutzerkennung verglichen, unter der sich der Benutzer angeloggt hat.

Im Anschluß an die Prüfsätze kann ein **Defaultsatz** angegeben werden. Im diesem Satz wird festgelegt, ob die Benutzung von CFS erlaubt ist oder nicht, falls weder die User-ID des Benutzers noch der Name der Datensichtstation in einem gewöhnlichen Prüfsatz gefunden wurde.

Der Default-Satz besitzt das Format: **DEFAULT=**Y | N

DEFAULT=Y bewirkt, daß CFS für die Benutzung zugelassen wird, falls in der CFS-Userfile kein Satz mit der User-ID oder dem Stationsnamen des Benutzers enthalten ist.

DEFAULT=N bewirkt, daß CFS für die Benutzung nicht zugelassen wird, falls in der CFS-Userfile kein Satz mit der User-ID oder dem Stationsnamen des Benutzers enthalten ist.

Standard: DEFAULT=Y

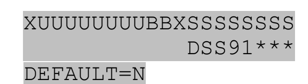
Beispiele:

```
XUUUUUUUUBBXSSSSSSSS
CTSOS        DSS4111
CTSOS        DSS4112
CTSOS        DSS4113
CTSOS       -
```

Unter der Kennung TSOS kann CFS nur an den Datensichtstationen DSS4711 bis DSS4713 aufgerufen werden. Von allen anderen Kennungen aus ist CFS ohne Einschränkungen benutzbar.

```
XUUUUUUUUBBXSSSSSSSS
-A10*****
```

Für alle Kennungen, die mit A10 beginnen, ist CFS nicht erlaubt. Für alle anderen Kennungen bestehen keine Einschränkungen.



```
XUUUUUUUUBBXSSSSSSSS
                DSS91***
DEFAULT=N
```

Die Benutzung von CFS ist nur an Datensichtgeräten erlaubt, deren Namen mit DSS91 beginnen.

Hinweis:

Die **Reihenfolge der Sätze** in der Userfile ist für das Ergebnis der Prüfung von Bedeutung. Siehe hierzu Beispiel 1 oben.

Beispiele für Userfiles finden Sie auch in der CFS.S.LMSLIB.

Hinweise zur Erstellung und Wartung der CFS-Userfile

- Um eine Userfile erstmalig zu erstellen, selektieren Sie mit Hilfe von CFS das Element USERFIL1 aus der CFS.S.LMSLIB: SEL USERFIL1,CFS.S.LMSLIB,CFSUSER . Der Skalierungssatz in der so erzeugten Datei CFSUSER dient als Orientierungshilfe für die von Ihnen neu einzutragenden Prüfsätze.

- Die Userfile muß shareable sein, so daß CFS von allen Benutzerkennungen darauf zugreifen kann.

- Die Userfile kann empfindliche Daten, wie z.B. Benutzerkennungen enthalten. Diese Informationen sollten nicht im Klartext für jedermann zugänglich sein. Aus diesem Grunde sind die in der Userfile enthaltenen Daten zu verschlüsseln. Von CFS werden nur die verschlüsselten Informationen verarbeitet.

- Der Vorgang des Verschlüsselns geschieht wie bei der Termfile mit Hilfe des Moduls CONCH. Die nähere Vorgehensweise ist im Abschnitt "Hinweise zur Erstellung und Wartung der Termfile" auf Seite A662 beschrieben.

- Aus Gründen der Datensicherheit sollte die Userfile vom CFS-Administrator auf ACCESS= READ gesetzt werden. In CFSMAIN kann außerdem ein READ-Passwort für die Userfile vereinbart werden (siehe Feld USERFPSW im Datenteil am Ende von CFSMAIN). Vor jedem Zugriff auf die Userfile wird dieses Passwort von CFS abgesetzt und nach erfolgreichem Open sofort wieder freigegeben. Das ursprüngliche CAT-Kommando für die Userfile muß der Systemverwalter selbst vornehmen: /CAT CFSUSER,STATE=U, RDPASS=......

- Die Durchführung der in diesem Abschnitt beschriebenen Berechtigungsprüfungen ist nur an das Vorhandensein der Userfile gebunden. Es sind weder Modifikationen im USRMOD vorzunehmen, noch sind irgendwelche Indikatoren im Modul CFSMAIN zu setzen. Es ist jedoch darauf zu achten, daß die Userfile den richtigen Dateinamen aufweist. Dieser kann im Datenteil des Moduls CFSMAIN definiert werden (Feld USERFILE).

6. Protokollierung der Benutzeraktivitäten

CFS bietet die Möglichkeit, alle CFS-Aufrufe zusammen mit Datum, Uhrzeit, TSN, Benutzerkennung, Abrechnungsnummer und Modus (Dialog/Enter) in einer Protokolldatei festzuhalten. Bei der Beendigung von CFS wird ebenfalls ein Protokollsatz mit der TSN, sowie der gesamten Verweilzeit in CFS in Minuten) ausgegeben.

Die Protokollierungsfunktion wird bei der Installation von CFS eingeschaltet, indem der Name der Protokolldatei von '*DUMMY' auf einen echten Dateinamen gesetzt wird. Dies kann geschehen durch eine Modifikation im 6. PAM-Block von CFS ab der Spalte 201. Näheres hierzu siehe Quellprogramm CFSMAIN in der CFS.S.LMSLIB. Die Protokolldatei (falls sie noch nicht existiert) wird von CFS shareable gemacht und mit einem Read-Passwort versehen.

Neben dem Aufruf von CFS können wahlweise die folgenden Benutzeraktivitäten protokolliert werden:

- Eröffnen einer Connection per OC-Kommando
- File-Transfer Anforderungen mittels Kommando/Action-Code FT oder Variabler Action ONXFT
- Modifizieren einer Datei im CFS-Editor.
- Fehlermeldungen des Connection-Handlers lassen sich für Testzwecke protokollieren.

Für eine Beschreibung der Aktivierung der verschiedenen Protokollierungen wird auf das Quellprogramm CFSMAIN in der CFS.S.LMSLIB verwiesen.

7. CFS.PDFILE für besondere Print-/FT-/RDAC-Kommandos

Mit der PDFILE wird dem CFS-Administrator und jedem einzelnen CFS-Benutzer die Möglichkeit geboten, gewissen alphanumerischen Mnemo-Codes (PD*mn*) bestimmte Remote-Drucker, Print-Kommandos, File-Transfer- oder RDAC-Profile zuzuordnen.

Die CFS.PDFILE ist eine gewöhnliche mit EDT erstellbare SAM-Datei, die 3 verschiedene Arten von Sätzen kennt:

- **Kommentarsätze** können an beliebiger Stelle eingestreut werden und beginnen in Spalte 1 mit einem Stern ('*').

- **Header-Sätze** beginnen in Spalte 1 mit '$'. Ab Spalte 2 folgt

  a) ein Gerüst von Parametern für das Kommando /PRINT (ISP).

  b) der String 'PRINT-FILE', gefolgt von einem Leerzeichen und Parametern für ein PRINT-FILE Kommando (SDF) oder
  der String 'PRINT-DOCUMENT', gefolgt von einem Leerzeichen und Parametern für ein PRINT-DOCUMENT Kommando (SDF).

  c) der String 'FT', gefolgt von einem Leerzeichen und Parametern für ein FT (File-Transfer) Kommando von CFS. Für die Beschreibung der Parameter siehe CFS-Benutzerhandbuch, Kapitel 7 (Kommando FT) und Kapitel 12 "File-Transfer mit FT-BS2000/RDAC".
  Beispiel: FT ,,,HOST1,USERID2,ACCNB2,C'PASS',,,,TO,YES.

  d) der String 'RDAC', gefolgt von einem Leerzeichen und Parametern für ein RDAC-Kommando von CFS (File-Transfer/Ausdruck mit RDAC). Für die Beschreibung der Parameter siehe CFS-Benutzerhandbuch, Kapitel 7 (Kommando RDAC) und Kapitel 12 "File-Transfer mit FT-BS2000/RDAC". Beispiel: RDAC ,,,DRUCKER2,,,,TO,SPACE=E,HEADER=YES.

  Im Fall a) und b) wird vorausgesetzt, daß der Dateiname der erste Parameter des Kommandos ist und zum Zeitpunkt des Ausdrucks einer Datei von CFS selbständig eingesetzt wird. Bei den Print-Parametern stehen die Sonderzeichen '*' stehen als Platzhalter für den Namen eines im PDN generierten Remote-Druckers.

  Im Fall c) und d) werden durch die ersten drei Kommas die Parameter *datei1*, *datei2*, *passw* übergangen. Der Parameter *datei1* wird zum Ausführungszeitpunkt durch den Namen der mit dem Action-Code PD markierten Datei versorgt. Für *datei2* wird der gleiche Name wie *datei1* eingesetzt.

  Ein Header-Satz kann bis zu 2048 Bytes lang sein. Falls die PDFILE jedoch mit EDT bearbeitbar sein soll, ist die maximale Satzlänge auf 256 Bytes begrenzt.

  Falls in einem Header-Satz der String '********' als Platzhalter für den Device-Namen angegeben wurde (z.B. DEV=********), so werden von CFS die nachfolgenden Mnemo-Device Zuordnungssätze (siehe unten) solange gelesen, bis ein Satz mit dem Mnemo-Code *mn* (entsprechend dem Action-Code PD*mn*) gefunden wird. Der in diesem Satz zugeordnete Drucker (Spalte 3 bis 10) wird dann an die Stelle von '********' im DEV- oder in sonstigen Print-Operanden eingefügt.

  Die Sonderzeichen '#' stehen für den im PD-Action-Code angegebenen Mnemo-Code. Falls in einem Header-Satz die Sonderzeichen '#' vorkommen, so wird an den entsprechenden Stellen der im PD-Action-Code angegebene Mnemo-Code direkt in die Parameter für das Print-Kommando übertragen. Für einen Header-Satz der die Zeichen '#' enthält, sind somit keine Mnemo-Device Zuordnungssätze (siehe unten) erforderlich.

Auf einen Header-Satz der Art a) bis d) folgen ein oder mehrere Mnemo-Zuordnungen (PD*mn*).

- **Mnemo-Zuordnungssätze** beginnen in Spalte 1 mit einem Zeichen ungleich '*' und '$'. Die Felder haben folgendes Format (siehe auch Beispiel auf der nächsten Seite):

  Spalte 1-2 Mnemonic *mn* des Action-Codes PD*mn* (*mn*: alphanumerisch, linksbündig). Der Mnemo-Code darf nicht mit den Zeichen B, E oder S enden.

  Spalte 3-10 DCAM Device-Name des Remote-Druckers / Name des Hostrechners für FT. Für den virtuellen Device-Namen ???????? wird beim ersten Ansprechen des zugehörigen Mnemo-Codes durch den Benutzer ein realer Device-Namen angefordert. Dieser Name bleibt während der CFS-Session solange gültig, bis er evtl. durch ein Kommando APD *device* geändert wird.

  Spalte 11-80 beliebiger Text (Beschreibung des Druckers, Standort, usw.) bzw. Blank. Die gesamte Liste aller Beschreibungen wird durch Angabe des Action-Codes PD? am Bildschirm ausgegeben.

  Falls in einem Header-Satz die **Platzhalterzeichen** '*' für den Device-Namen **nicht** angegeben wurden (z.B. CONTROL=PHYS,FORM=STD,LOOP=C8,CHARS=(82E,82R)), so ist der Device-Name in dem nachfolgenden Mnemo-Device Satz nur aus formellen Gründen erforderlich. In diesem Fall ist lediglich der Mnemo-Code von Bedeutung. Über diesen wird die Zuordnung zu dem Print-Kommando mit den im vorausgehenden Header-Satz angegebenen Parametern hergestellt.

Zuordnungsalgorithmus Mnemo-Code <--> durch CFS erzeugtes Print-/FT-/RDAC-Kommando:

Action-Code PD*mn*: Es wird die Bezeichnung *mn* in den Spalten 1 bis 2 der PDFILE gesucht.

Falls ein Satz mit der angegebenen Bezeichnung gefunden wurde, wird der vorausgegangene Header-Satz ($...) zur Ermittlung des auszuführenden Kommandos gesucht.

Alle in den Print-Parametern des Header-Satzes vorkommenden '*'-Zeichen werden sukzessive durch die entsprechenden Stellen des Device-Namens ersetzt, sofern der Device-Name nicht auf ???????? lautet (virtueller Device-Name, siehe oben). In dem so gebildeten Print-Kommando ist lediglich noch der Name der Datei offen. Dieser wird von CFS vor der Ausführung des Kommandos eingefügt.

Falls kein Satz mit dem angegebenen Mnemo-Code gefunden wurde, so wird der erste Header-Satz gesucht, in dem mindestens ein '#' vorkommt. Wird ein derartiger Header-Satz gefunden, so wird der im PD-Action-Code angegebene Mnemo-Code sukzessive für jedes der 1 bis 2 '#'-Zeichen substituiert. Dabei ist folgendes zu beachten: Ist im PD-Action-Code ein 1-stelliger Mnemo-Code angegeben (PD5) und im Header-Satz wird ein 2-stelliger Mnemo-Code erwartet (##), so wird der PD-Mnemo-Code intern auf 2 Stellen erweitert (PD05).

Trifft keiner der oben beschriebenen Fälle zu, so wird am Bildschirm eine Fehlermeldung ausgegeben "Header-Record/Station-Record missing in PDFILE".

Mit Hilfe des Action-Codes PD (bzw. PD?) kann sich der Benutzer eine Liste aller installierten Mnemo-Codes und deren Zuordnung zu den entsprechenden Print-Kommandos, bzw. Remote-Druckern ausgeben lassen.

Der Name der PDFILE und die Kennung kann vom CFS-Administrator im Patch-Bereich von CFS frei bestimmt werden (siehe Quellprogramm CFSMAIN).

Eine zentrale, von allen Benutzern zu verwendende CFS.PDFILE muß shareable sein. Es kann sich jeder Benutzer aber auch seine eigene PDFILE erstellen und per FILE-Kommando zuweisen: /FILE *pdfile*,LINK=PDFILE.

```
Beispiel für eine PDFILE:
```

```
$CONTROL=PHYS,FORM=SX1,LOOP=C8,CHARS=(82E,82R)
Z1        zentraler Laserdrucker; Format 1 (siehe RZ-Spezifikation)
$DEV=********,FORM=H810
A0DRU1    Tischlaser Raum 418, Format: 8 LPI, Schrift: 10 CPI
..............

H0DRU8               Raum 502, Format: 8 LPI, Schrift: 10 CPI
$PRINT-FILE ALT-USER=*NONE,DEVICE-NAME=********,START-OUTPUT=STD
A3DRU1    Tischlaser Raum 418, Format: 8 LPI, Schrift: 12 CPI
B2DRU2               Raum 422, Format: 8 LPI, Schrift: 12 CPI
..............

X ????????           virtueller Devicename zum Mnemo-Code X.
$FT ,,,********,USERID0,ACCOUNT0,C'pass',/DO succ-proc,/DO fail-proc,*BS2000,TO,YES
F0HOST0   File-Transfer nach HOST0, User-Id USERID0
F1HOST1   File-Transfer nach HOST1, User-Id USERID0
$RDAC ,,,********,,,,TO,YES,SPACE=E,HEADER=HDR1
R0DRUCK10 RDAC-Drucker 10
R1DRUCK11 RDAC-Drucker 11
..<3 - 10><11-----------------------------------------------------------------2048>
 |    |        |
 |    |        ▫--- Beliebiger Kommentar. Dieser wird bei Eingabe
 |    |             des Action-Codes  PD / PD?  für alle definierten
 |    |             Drucker angezeigt.
 |    |
 |    ▫- Device-Name des Druckers/Host-Rechners.
 |
 ▫-- 2 Stellen Mnemo-Code:  Beschreibung des Druckers und des besonderen Formats.
```

8. Installation von CFS

Die Installation von CFS erfolgt durch Einspielen eines ARCHIVE-Bandes unter einer beliebigen Benutzerkennung, z.B. unter $CFS. Folgende Anweisungen sind hierzu notwendig:

```
/EXEC $ARCHIVE
F NA=($TT08.,RENAME=$CFS.)
I FROM=(CFSxxx),DEVICE=xxxx,L=SYSOUT,REP=ALL
END
```

Die Dateien CFS, CFSLIB und CFSHELP sollten auf die Eigenschaft SHARE=YES überprüft werden.

In der mit dem Magnetband ausgelieferten Version sind die Standardwerte der CFS-Kommandos und Optionen so eingestellt wie in Kapitel 18 "Parameter ändern" beschrieben.

Hiervon abweichende Standardeinstellungen können realisiert werden auf Benutzerebene mit Hilfe von Startup-Dateien. Startup-Dateien enthalten Kommandos, die nach Aufruf des Programms automatisch ausgeführt werden. Jeder Benutzer kann hier seine individuellen Parameter-Voreinstellungen festlegen.

Eine **RZ-globale Änderung der Standardwerte** wird erreicht durch das Vorschaltprogramm CFSMAIN. Dieses in Assembler-Sprache geschriebene Programm wird im Source-Code ausgeliefert und steht unter dem Namen CFSMAIN in der CFS.S.LMSLIB zur Verfügung. Über CFSMAIN können Parameter-Voreinstellungen in CFS geändert, sowie Datenschutzeinrichtungen aktiviert werden.

Beispiele:

- Aktivierung des Prüfmoduls USRMOD. Nach jeder Eingabe wird dieser Modul durchlaufen, wobei die Benutzereingaben geprüft bzw. modifiziert werden können.
- Bestätigung durch den Benutzer vor dem Überschreiben von bestehenden Dateien, z.B. bei Select, Copy, Add, Erase.
- Maximale Anzahl der zu eröffnenden Connections.
- Zeichen für das Kommandogedächtnis in Connections (Standard: '-').
- Standard User-Option für die Dateiselektion.
- Namen von CFS-internen Dateien. Z.B. Name einer Protokollierungsdatei, in der alle CFS-Aufrufe durch die Benutzer festgehalten werden können.
- Namen der OMLs für Standard-Softwareprodukte, die von CFS unterstützt werden.
  z.B. OMLPERCON, LMSLIB, FMSLIB, FMS.LNKLIB, SYSOML.FLAM, EDTLIB, TOM.TI.LNKLIB.

Im Quellprogramm CFSMAIN sind alle derzeit realisierten Änderungsmöglichkeiten zusammen mit einer kurzen Beschreibung aufgeführt.

Zur Änderung einer Option auf einen vom Standard abweichenden Wert genügt es in der Regel, den Kommentarstern aus dem Source-Code zu entfernen und damit den entsprechenden MVI-/ MVC-Befehl zu aktivieren.

Das vom Systemverwalter modifizierte Programm CFSMAIN ist mit der Prozedur ASSEMB aus der CFS.S.LMSLIB zu übersetzen. Die angegebenen Makrobibliotheken in dieser Prozedur sollten zuvor überprüft und ggf. korrigiert werden. Der Aufrufparameter &PLAM der Prozedur ASSEMB ist standardmäßig auf ON gesetzt, was bedeutet, daß die CFSLIB im Format einer PLAM-Bibliothek vorliegt.

Damit die gewünschten Modifikationen in CFS wirksam werden, muß das Programm CFS neu gebunden werden. Hierfür sind die Prozeduren CFSLNK und CFSLNK1 in der CFS.S.LMSLIB vorgesehen. Im Normalfall ist die Prozedur CFSLNK zu verwenden. Die Prozedur CFSLNK1 sollte nur benutzt werden, wenn Nachlademodule aus der CFSLIB wie MODR oder USRMOD fest in CFS eingebunden werden. Hierzu ist auch eine Modifikation in CFSMAIN notwendig.

Bei einer **späteren Auslieferung von CFS** können die oben dargestellten Schritte abgekürzt werden bzw. entfallen, sofern folgende Vorgehensweise eingehalten wird:

Die bisherigen Dateien CFSLIB und CFS.S.LMSLIB in Sicherungsdateien kopieren:

/COPY CFSLIB,CFSLIB.ALT
/COPY CFS.S.LMSLIB,CFS.S.LMSLIB.ALT.

Die alten Versionen dieser Dateien stehen damit auch nach dem Einspielen der neue CFS-Version noch zur Verfügung.

Die Module CFSMAIN, USRMOD, MODR und eventuell weitere benutzerspezifische Module sind aus der alten CFSLIB in die neu ausgelieferte CFSLIB zu übertragen. CFS ist anschießend neu zu binden (DO CFS.S.LMSLIB(CFSLNK/CFSLNK1)).

Die Adressen der CFS-Parameter sind aufwärtskompatibel zu zukünftigen Versionen von CFS. Aus diesem Grunde kann auch der alte Modul CFSMAIN mit einer neuen CFS-Programmversion zusammengebunden werden. Im aktuellen Quellprogramm CFSMAIN werden in der Regel zusätzliche Modifikationsmöglichkeiten enthalten sein, die es in der alten CFS-Version noch nicht gab. Aus diesem Grunde empfiehlt es sich, das alte Quellprogramm von CFSMAIN mit der neu ausgelieferten zu vergleichen (EDT-Compare), die im alten CFSMAIN vorgenommenen Modifikationen in das neue Quellprogramm zu übertragen und die zusätzlichen Möglichkeiten zur Modifikation von Parametern entsprechend den RZ-Bedürfnissen zu aktivieren.

Für die Namen der **CFS-internen Dateien**, die standardmäßig auf die Namen $TSOS.CFS*xxxx* voreingestellt sind, gilt folgendes: Über den Linknamen ECERDLOD wird die Benutzerkennung ermittelt, von der das Programm CFS geladen wurde (z.B. $CFS.). Ist diese Kennung verschieden von TSOS, so wird in den internen Dateinamen die Standard-Benutzerkennung $TSOS. gegen diejenige Kennung ausgetauscht, unter der CFS geladen wurde (z.B. $CFS.). Der Austausch der Benutzerkennung wird nicht durchgeführt, falls die User-Id eines internen Dateinamens bereits auf einen Wert ungleich TSOS geändert wurde (siehe entsprechende MVC-Befehle im Initialisierungsteil von CFSMAIN).

Der Systemverwalter kann verschiedene Versionen des Produkts CFS zur Benutzung freigeben. Hierzu sind die Namen der Dateien CFS, CFSLIB, CFSHELP und evtl. die Jobvariable CFS.MESSAGE mit einer Versionssuffix .*nnn* bzw. .V*nnn* zu versehen (z.B. CFS.945, CFSLIB.945, CFSHELP.945). Die so erzeugten neuen Dateinamen müssen in CFSMAIN nicht eigens bekannt gemacht werden. Die Namen werden nach einem ähnlichen Automatismus wie oben beschrieben mit den korrekten Versionsbezeichnungen ergänzt.

Falls CFS auf PC's mit 9750-**Terminalemulationen** betrieben wird, so ist es u.U. notwendig, vor Aufruf des Programms den Prozeßschalter 2 zu setzen. Das Setzen dieses Prozeßschalters kann auch über CFSMAIN automatisch vorgenommen werden.

CFS-Bulletin

Der Systemverwalter hat die Möglichkeit, in einer Jobvariablen einen bis zu 240 Byte langen Text für ein **Bulletin an alle CFS-Benutzer** zu hinterlegen. Dieser Text (z.B. "Neues CFS im Einsatz. Die Änderungen gegenüber der bisherigen Version erhalten Sie, indem Sie im Feld FILENAME-SELECT ?all eingeben und im nachfolgenden Menue den ersten Punkt ankreuzen.") wird bei jedem Aufruf von CFS im unteren Teil der Selektionsmaske einmal angezeigt. Bei der zweiten und allen folgenden Ausgaben der Selektionsmaske innerhalb des gleichen CFS-Laufs wird der Bulletintext nicht angezeigt. Die Jobvariable für den Text des Bulletins hat den Namen CFS.MESSAGE und muß unter der Kennung eingerichtet werden, unter der auch CFS gespeichert ist, z.B. $CFS Die Jobvariable muß außerdem shareable sein. Nach gegebener Zeit kann die Jobvariable vom Systemverwalter nonshareable gemacht, oder ganz gelöscht werden.

Benutzereigene FSTAT-Routine zum schnellen Kataloglesen

Der Systemverwalter hat die Möglichkeit, zum Zwecke einer beschleunigten Katalogverarbeitung eine eigene FSTAT-Routine in CFS zu integrieren. Diese Routine erhält als Eingabeparameter in Register 1 die Adresse der Parameterliste für einen FSTAT-Makroaufruf von der Art FSTAT *datei*,*bereich*,*länge*, LONG,MF=L. Aufgrund der in dieser Parameterliste abgelegten Parameter ist der Katalogeintrag für die gewünschte Datei in der gewünschten Länge im angegebenen Bereich bereitzustellen. Dieses Bereitstellen des Katalogeintrags kann durch direktes Kataloglesen mit Hilfe einer P2-Routine oder durch Ausführen eines FSTAT-Makros geschehen (FSTAT MF=(E,(1)).

Aufrufkonventionen:

R1	Adresse der Parameterliste
R13	Adresse einer 72 Byte langen Save-Area
R14	Returnadresse in CFS
R15	Ansprungadresse des Benutzermoduls

Die **Registersicherstellung** im Benutzermodul sollte nach den Standardkonventionen erfolgen:

```
STM   14,12,12(13)          (Ansprung)
...
LM    14,12,12(13)          (Rückkehr)
```

Returncodes:

Im Register 15 ist beim Verlassen der Routine ein Returncode an CFs zurückzugeben.

R15 = 0: Katalogeintrag für die gewünschte Datei wurde erfolgreich bereitgestellt.

R15 > 0: Katalogeintrag konnte für die gewünschte Datei nicht bereitgestellt werden.
R15 kann in diesem Fall z.B. mit dem DMS-Fehlercode gefüllt werden.

Die Einbindung der benutzereigenen FSTAT-Routine erfolgt über CFSMAIN durch Entfernen des Kommentarsterns vor der MVC-Anweisung, die die V-Adresse der Routine in den globalen CFS-Datenbereich überträgt. Durch Binden von CFS mit Hilfe der DO-Prozedur CFS.S.LMSLIB (CFSMAIN) wird die Benutzerroutine über Autolink oder durch eine explizite Include-Anweisung (INCLUDE FSTAT [, *bibliothek*]) in CFS integriert.

Hinweis: Die FSTAT-Routine wird, falls aktiviert, **stets** zum Lesen der Katalogeinträge für die zu selektierenden Dateien verwendet. Dies gilt auch in den Fällen, in denen das Programm CFS unter Benutzerkennungen, die keine Privilegierung besitzen, geladen wurde.

9. Ausgelieferte Dateien

CFS	Gebundene Programmphase.
CFS.S.LMSLIB	LMS-Bibliothek im PLAM-Format mit CFS Quellprogrammen, Prozeduren und Beispielen. Eine Beschreibung des Inhalts dieser Bibliothek folgt weiter unten.
CFS.SYSTEM.SYMBOLS	Hilfsdatei zur symbolischen Anzeige verschiedener Systemtabellen (z.B. TCB/XVT) unter der CFS-Komponente TAS (Task Services).
CFS.README	Diese Datei enthält Hinweise zum Thema "CFS als Subsystem".
CFSHELP	Texte für die Helpfunktion von CFS.
CFSHELP-E	Texte für die Helpfunktion von CFS in englischer Sprache. Bei Bedarf muß diese Datei in CFSHELP umbenannt werden.
CFSLIB	Nachladebibliothek für CFS (PLAM-Format).
SYSSSD.CFS.xxx	CFS-Subsystemdeklaration für verschiedene BS2000/OSD-Versionen.

Inhalt der Bibliothek CFS.S.LMSLIB

J/ASSEMB	Übersetzungsprozedur für die in der Bibliothek gespeicherten Quellprogramme CFSMAIN, USRMOD, TEST, PREMOTE usw. Die Namen der Makrobibliotheken sind auf BS2000 Version 9.5 ausgerichtet. Als Übersetzungsprogramm wird ASSEMBF verwendet.
J/ASSEMB10	Wie oben. Die Namen der Makrobibliotheken sind jedoch auf BS2000 Version 10.0 ausgerichtet. Als Übersetzungsprogramm wird ASSEMBF verwendet.
J/ASSEMB11	Wie oben. Die Namen der Makrobibliotheken sind jedoch auf OSD V1 ausgerichtet. Als Übersetzungsprogramm wird ASSEMBH verwendet.
J/CFS.DO.CONV.CFSLIB	Prozedur zur Umsetzung der CFSLIB in eine OML (LMR-Format).
J/CFS.DO.MIX.SDF	Prozedur zum Einfügen der mitgelieferten Syntax-Datei CFS.SYNTAX in die zentrale Syntaxdatei von SDF. Die Prozedur setzt das Programm SDF-I voraus. Nach fehlerfreiem Ablauf der Prozedur sind die BS2000-Kommandos CFS und NP implementiert. Diese können nach dem nächsten Systemstart benutzt werden. Eine funktionelle Beschreibung der Kommandos finden Sie in der DO-Prozedur CFS.DO. START.
J/CFS.DO.PRINT	DO-Prozedur zum Ausdrucken laseraufbereiteter CFS-Hardcopies. Zugehörig hierzu: CFS.ND-/HPFILE. Der Systemverwalter muß die PROC-Anweisung dieser Prozedur evtl. noch bezüglich der verwendeten HP-/NDFILE, sowie bezüglich der verwendeten Zeichensätze anpassen. Hinweise hierzu sind in der Prozedurdatei enthalten.
J/CFS.DO.START	Diese Prozedur wird benötigt, falls das SDF-Kommando CFS im BS2000-System implementiert wurde (siehe hierzu auch die Dateien CFS.DO.SDF und CFS.SYNTAX). Nach Eingabe des Kommandos CFS wird diese DO-Prozedur zum Starten des Programms aufgerufen.
J/CFSHT	CFS-Holdertask. Diese Enter-Datei muß vom Systemverwalter unter TSOS gestartet werden. Der Holdertask ermöglicht es Nicht-TSOS Benutzern, die User Option OPEN zu verwenden. Dem Benutzer werden in einer Liste alle Dateien angezeigt, die von Tasks eröffnet sind, welche unter der eigenen Kennung laufen. Der Holdertask ermöglicht es ferner, den momentanen Inhalt von sequentiellen Output-Dateien mit dem Action-Code D sichtbar zu machen. Außerdem können bestimmte TSOS-Kommandos dem nichtprivilegierten Benutzer zur Verfügung gestellt werden. Siehe hierzu auch die Beschreibung zum Element X/CFS.HTCMD weiter unten.
J/CFSLNK	CFS-Binderprozedur. Hiermit wird aus den Modulen CFSMAIN und CFSUP eine ablauffähige Programmphase CFS gebunden.
J/CFSLNK1	CFS-Binderprozedur. Es werden zusätzliche Module wie z.B. MODR, USRMOD eingebunden, die sonst über LINK nachgeladen werden.
J/CFSLNK2	Prozedur zum Binden eines Großmoduls CFS. Weitere Informationen siehe CFS-Benutzerhandbuch, Kapitel 17 ("CFS als Unterprogramm").
J/GENHELP	Prozedur zum Erstellen einer benutzereigenen Help-Datei CFSHELP. USER . Der Benutzer kann auf die Texte in dieser Help-Datei durch das Kommando ?USER zugreifen.

J/JESLNK	Prozedur zum Binden eines Großmoduls JES aus dem benutzerspezifischen Vorlaufmodul JESMAIN und dem JES-Hauptmodul. Diese Prozedur ist nach jeder Änderung von JESMAIN aufzurufen.
J/REP	Prozedur Eintragen des CPU-Reps in den aktuellen Modul MODR.
J/TASHT	TAS-Holdertask. Diese Enter-Datei muß vom Systemverwalter unter TSOS gestartet werden. Der Holdertask ermöglicht es Nicht-TSOS Benutzern, eine eingeschränkte Auswahl der von der Komponente TAS angebotenen Funktionen zu nutzen. Die Auswahl der dem Benutzer zugänglichen Funktionen wird durch die maximale Testprivilegierung seiner Benutzerkennung geregelt. Für eine ausführliche Beschreibung wird auf das Quellprogramm S/TASMAIN in der CFS.S.LMSLIB verwiesen.
M/...	Makros, die zum Übersetzen der mitgelieferten Quellprogramme (CFSMAIN, usw.) benötigt werden.
S/CEXIT0A	Quellprogramm für Connection-Exit zur Prüfung der Benutzereingaben. Bezüglich der Aufrufparameter siehe entsprechende Inline-Dokumentation.
S/CFSHELP.HELPTABS	Copy-Daten zur Generierung der CFSHELP.USER-Datei. Die Daten sind als Beispiel gedacht und können beliebig ergänzt oder durch andere Texte ersetzt werden.
S/CFSHELP.USER.MENUE	Daten zur Generierung einer benutzereigenen Helpdatei CFSHELP. USER. Die Daten sind als Beispiel gedacht und können beliebig ergänzt oder durch andere Texte ersetzt werden. Das vorliegende Datenelement zeigt den Aufbau von Menues mit Verweisen auf die zugehörigen Textpassagen.
S/CFSMAIN	Quellprogramm des Vorschaltmoduls für CFS. Im Quellprogramm selbst sind die Möglichkeiten der Änderung von CFS-Standardeinstellungen dokumentiert. Weitere Informationen siehe A 670.
S/CONV	Quellprogramm für benutzereigene Variable Actions. Das Programm realisiert die Variable Action ONXCONV. Benutzereigene Variable Actions müssen in Klammern angegeben werden (z.B. ONX(CONV)). Weiterführende Informationen zu Benutzerexits finden Sie in Kapitel 17 des CFS-Benutzerhandbuchs.
S/FREE	Quellprogramm für benutzereigene User Options. Das Programm realisiert die User Option FREE. Benutzereigene User Options müssen in Klammern angegeben werden. Weitere Informationen zu Benutzerexits finden Sie in Kapitel 17 des CFS-Benutzerhandbuchs.
S/IFUQ	Quellprogramm des gleichnamigen Run-Moduls für die CFS-Prozedursprache (*RUN IFUQ ....). Die Funktionen des Moduls sind im Quellprogramm beschrieben.
S/JESEXIT...	Quellprogramme aller existierenden Benutzerexits in JES. Die entsprechenden Schnittstellen sind in einem eigenen Dokument beschrieben.
S/JESMAIN	Quellprogramm des Vorschaltmoduls für die Komponente JES. Im Quellprogramm selbst sind die Möglichkeiten der Änderung von JES-Standardeinstellungen dokumentiert. Weitere Informationen finden Sie auch am Ende des Manuals JES-Benutzerbeschreibung.

S/PREMOTE	Quellprogramm für einen benutzereigenen Action-Code. Das Programm realisiert den Action-Code PDxx. Benutzereigene Action-Codes müssen in einer internen Tabelle deklariert werden. Näheres hierzu siehe Programm CFSMAIN. Für eine Beschreibung der Funktionalität von PREMOTE siehe A **667**. Weiterführende Informationen zu Benutzerexits finden Sie in Kapitel 17 des CFS-Benutzerhandbuchs.
S/SETLEN	Quellprogramm des gleichnamigen Run-Moduls für die CFS-Prozedursprache (*RUN SETLEN ....). Die Funktionen des Moduls sind im Quellprogramm beschrieben.
S/SVAR	Quellprogramm des gleichnamigen Run-Moduls für die CFS-Prozedursprache (*RUN SVAR ....). Die Funktionen des Moduls sind im Quellprogramm beschrieben.
S/TASMAIN	Quellprogramm des Vorschaltmoduls für die Komponente TAS. Im Quellprogramm selbst sind die Möglichkeiten der Änderung von TAS-Standardeinstellungen dokumentiert.
S/TEST	Beispiel für ein Hauptprogramm, das CFS als Unterprogramm aufruft und bestimmte Funktionen nutzt. Weiterführende Informationen hierzu finden Sie in Kapitel 17 des CFS-Benutzerhandbuchs ("CFS als Unterprogramm").
S/USRMOD	Quellprogramm des Moduls zur Aktivierung von Datenschutz- und Berechtigungsprüfungen bei einzelnen CFS-Funktionen. Eine ausführliche Beschreibung des Vorgehens finden Sie auf Seite A **649**.
VSNSP	Quellprogramm für das CFS-Kommando (VSNSP). Benutzereigene Kommandos müssen in Klammern angegeben werden.
X/CFS.HPFILE	Zeichensatz zum Ausdruck von laseraufbereiteten CFS-Hardcopies an 3352-Laserdruckern (HP-Druckern).
X/CFS.HPFILE.027	Zeichensatz zum Ausdruck von laseraufbereiteten CFS-Hardcopies an 3352-Laserdruckern (HP-Druckern). Für SPOOL Version 2.7.
X/CFS.HTCMD	Diese Datei enthält die Angaben um privilegierte TSOS-Kommandos nicht-TSOS Benutzern zur Verfügung zu stellen. HTCMD ist eine Hilfsdatei zu J/CFSHT (CFS-Holdertask).
X/CFS.MENUE	Beispiel für eine Menue-Datei. Näheres siehe Kapitel 10 des CFS-Benutzerhandbuchs ("Menue-System", Abschnitt "Format der Menue-Datei").
X/CFS.MENUE.FORMAT.x	Beispiel für Maskendateien des Menue-Systems für Connections. Näheres siehe Kapitel 10 des CFS-Benutzerhandbuchs ("Menue-System für Connections", Abschnitt "Benutzerspezifische Gestaltung der Menue-Maske").
X/CFS.NDFILE	Zeichensatz zum Ausdruck von laseraufbereiteten CFS-Hardcopies an 3350-Laserdruckern (ND-Druckern).
X/CFS.PDFILE	Beispiel für eine PDFILE. Näheres siehe A **667** bzw. Beschreibung des Elements S/PREMOTE weiter oben.
X/CFS.SYNTAX	Mit SDF-A erzeugte Syntax-Datei für die BS2000-Kommandos CFS und NP. In der Syntax-Datei ist der Name der DO-Prozedur zum Starten von CFS eingetragen (Standard: $CFS.CFS.DO.START). Die Benutzerkennung der Prozedur muß u.U. korrigiert werden.

X/CFS.SYSTEM.SYMBOLS	Datei zur symbolischen Anzeige bestimmter Systemtabellen wie TCB und XVT durch das SHOW-Kommando der Komponente TAS (Task Services).
X/CFS.USERLIB.BEISPIEL	PLAM-Bibliothek mit interessanten **Anwendungsbeispielen für CFS-Prozeduren**.
X/CFS.USERACT.BEISPIEL	Datei zur Definition benutzereigener Action-Codes (%*xy*). Für weitere Informationen siehe CFS-Benutzerhandbuch, Kapitel 6. Nach dem Selektieren auf Platte sollte die Datei in CFS.USERACT umbenannt und shareable gemacht werden.

X/CFSHELP*x*.EXE	Selbstextrahierende Archive mit den Windows-Hilfe Dateien zu CFS/TAS/JES/JESG. Die Dateien sind auf Platte zu selektieren und anschließend mit FTP im Binärformat zu einen DOS/Windows 3.1/Win 95/Win NT-Rechner zu transferieren. Nach korrekter Übertragung können die Dateien auf dem DOS-Prompt durch Aufruf des Kommandos CFSHELPA bzw. CFSHELPB entpackt werden. Dies sollte in einem Verzeichnis *x*:*abc* erfolgen, in dem die CFS Winhelp-Dateien letztendlich stehen sollen. Anschließend sind unter Windows zwei Icons mit folgender Befehlszeile zu erstellen:
	`winhelp x:\abc\cfs.hlp`
	`winhelp x:\abc\info.hlp`

X/CFSHELP-E	Texte für die Helpfunktion von CFS in englischer Sprache. Bei Bedarf auf Platte selektieren, shareable machen und in CFSHELP umbenennen.
X/CFSTERM.0	Leerformular für CFS-Termfiles. Über eine Termfile kann geprüft werden, ob Benutzer bestimmte Connections eröffnen dürfen. Für ausführliche Informationen siehe A **656**.
X/CFSUSER.1	Beispiel für eine CFS-Userfile. Über eine Userfile können bestimmte User-ID's /Bildschirme von der Benutzung des CFS ausgeschlossen werden. Für ausführliche Informationen siehe A **663**.
X/CFS.SYSSSD..	Subsystemdeklarationen zu CFS für die verschiedenen BS2000/OSD-Versionen.